INTRODUCTION AUX RÉSEAUX DE NEURONES

Apprentissage de représentations et applications aux données textuelles

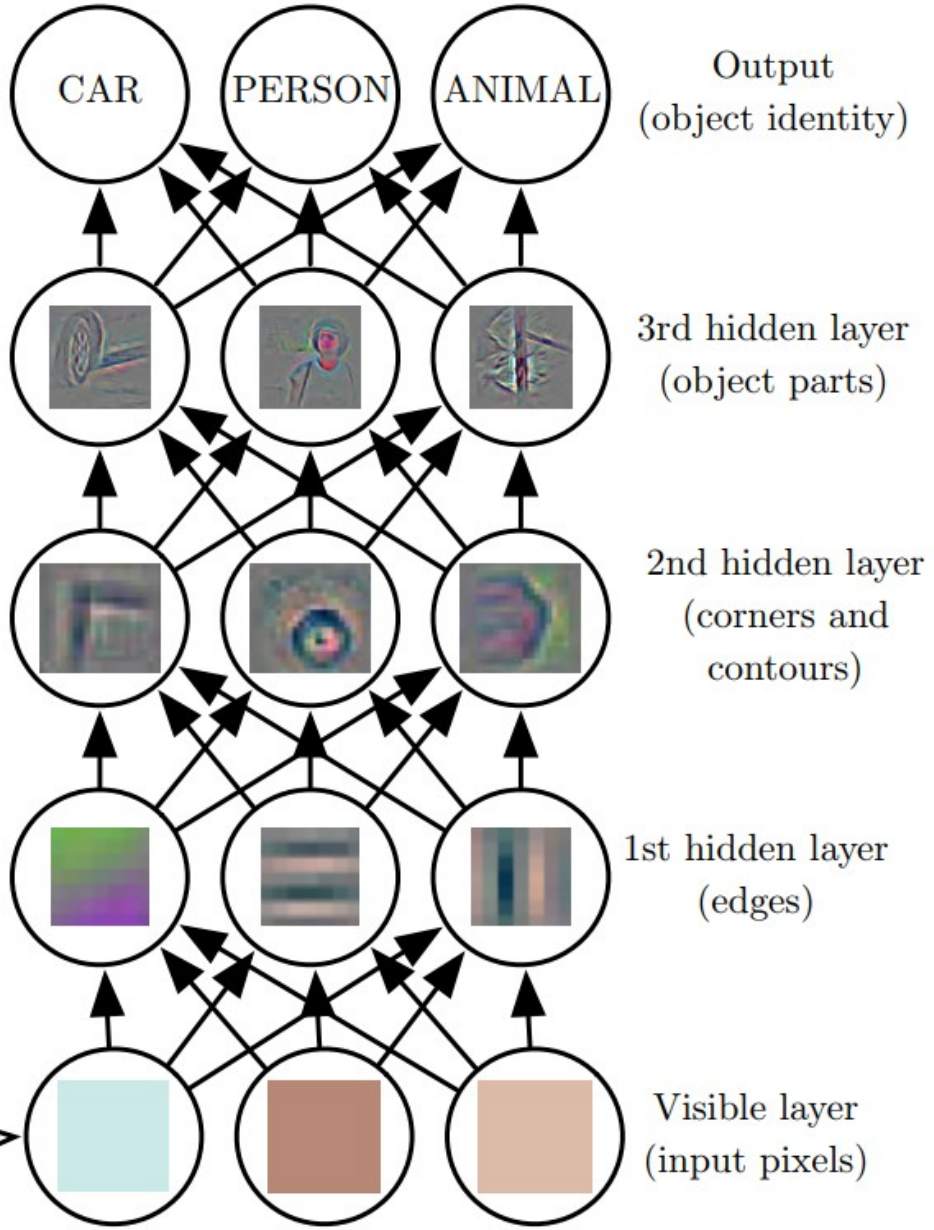
Pascal Germain*, 2018

* Merci spécial à Philippe Giguère pour m'avoir permis de réutiliser une partie de ces transparents.

Transfert et apprentissage supervisé

Hiérarchie de filtres

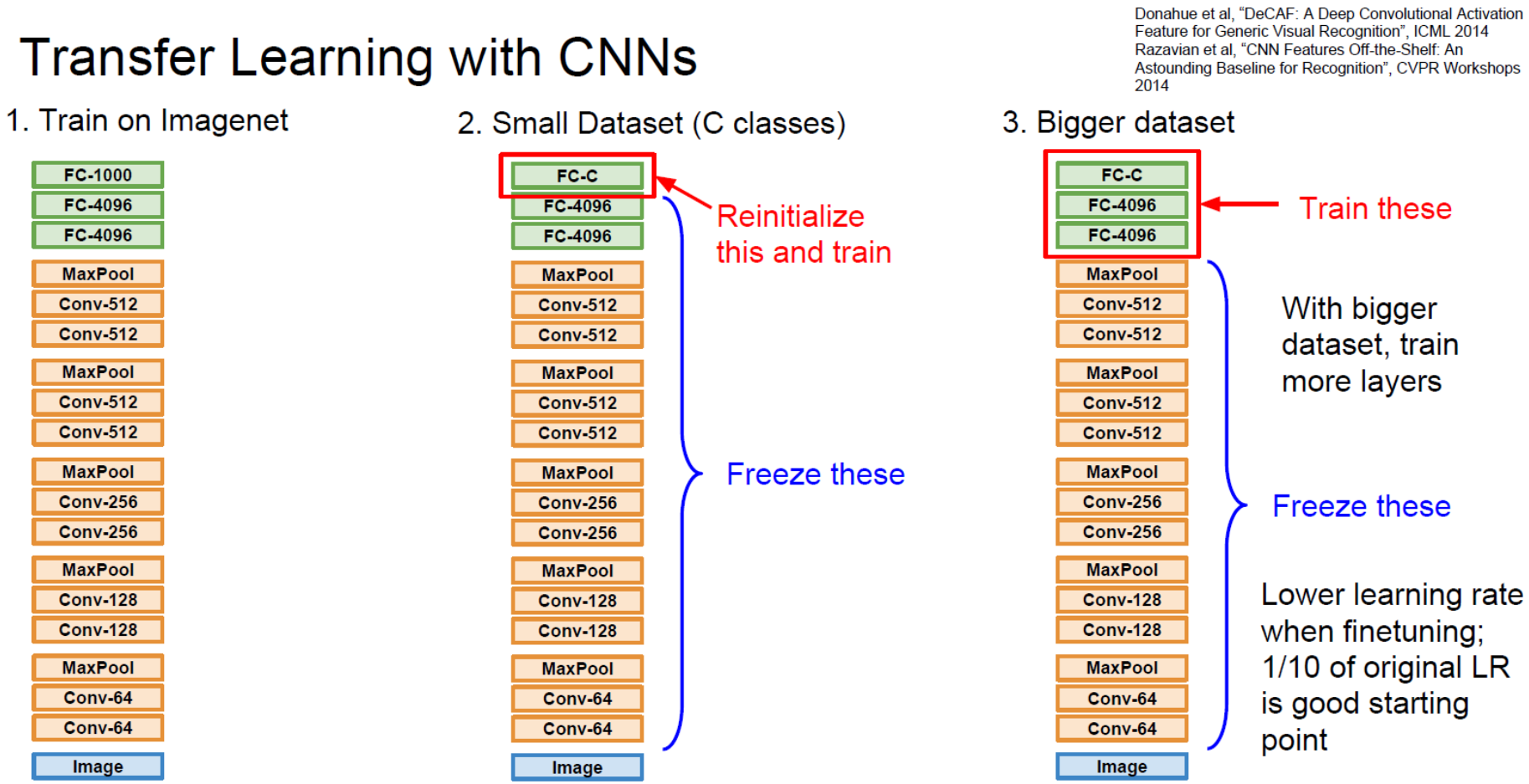
Va établir des liens entre des pixels de plus en plus éloignés



(I. Goodfellow) deeplearningbook.org

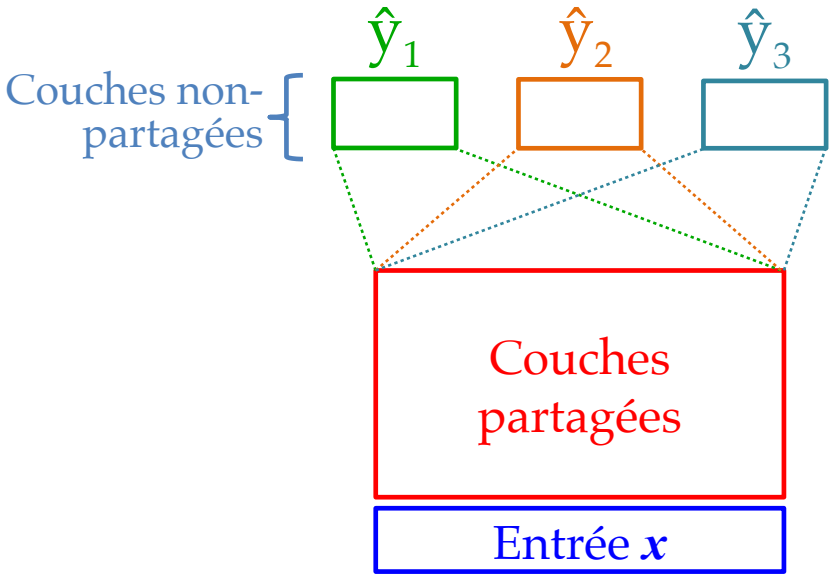
« Finetuning »

Permet le transfert de l'apprentissage vers une tâche similaire



tiré de : cs231n

Apprentissage multitâche



- Doit y avoir un certain lien entre les tâches

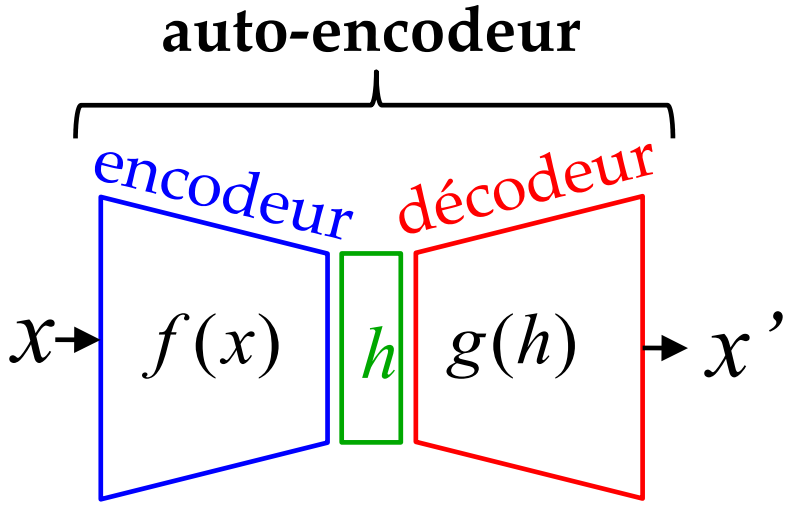
Auto-encodeur

Perte de reconstruction

- **Supervisé** : Typiquement, la perte est basée sur l'erreur entre prédiction et la réponse désirée.

Exemple : $L_{quad}(R(x), y) = (R(x) - y)^2$

- **Non-supervisé** : Un **autoencodeur** minimise une perte basée sur la reconstruction de l'entrée x

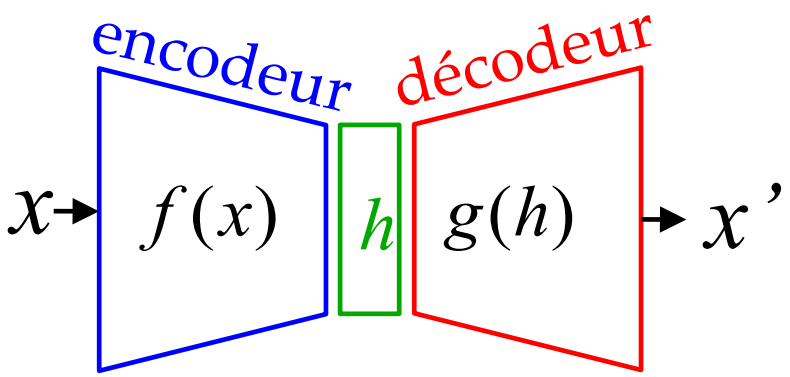


Pour éviter des solutions inintéressantes

Perte : $L = \left| g(f(x)) - x \right|^2$ + régularisation

Taxonomie

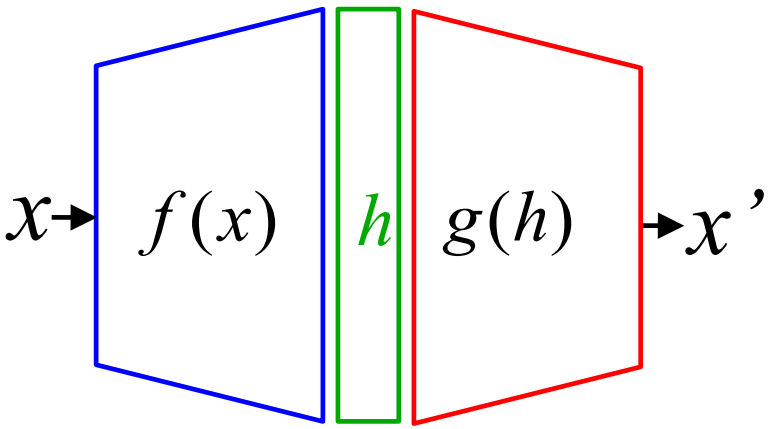
«Undercomplete»



taille x > taille h

- L'encodeur doit trouver une projection vers un espace de plus petite dimension
- si f, g sont linéaire:
  proche de l'ACP
  (exactement l'ACP avec
  $f=U^T$, $g = U$, $U^TU=I$)

«Overcomplete»



taille x < taille h

- sera inutile sans régularisation :
  h peut contenir une copie de x
- exemple de régularisation :
  x bruité

Importance de la régularisation

- Sans régularisation, l'encodage pourrait être inutile
  - Cas pathologique théorique : encodeur-décodeur très puissant, taille $h = 1$

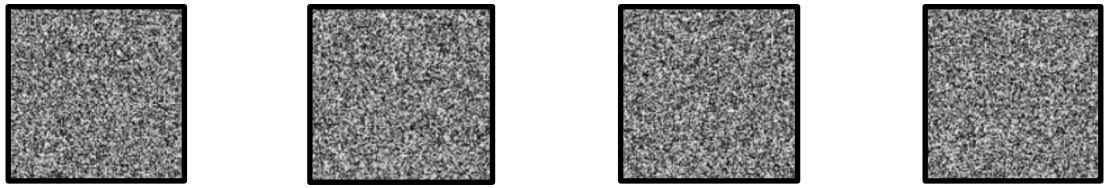
$$x \rightarrow \text{indice } i \rightarrow x$$

(certains réseaux profonds peuvent apprendre par cœur des jeux de données)

Variété (*manifold*)

- La plupart des données réelles vont résider dans des sous-régions particulières de l'espace de x

pixels tirés au hasard : uniforme dans x

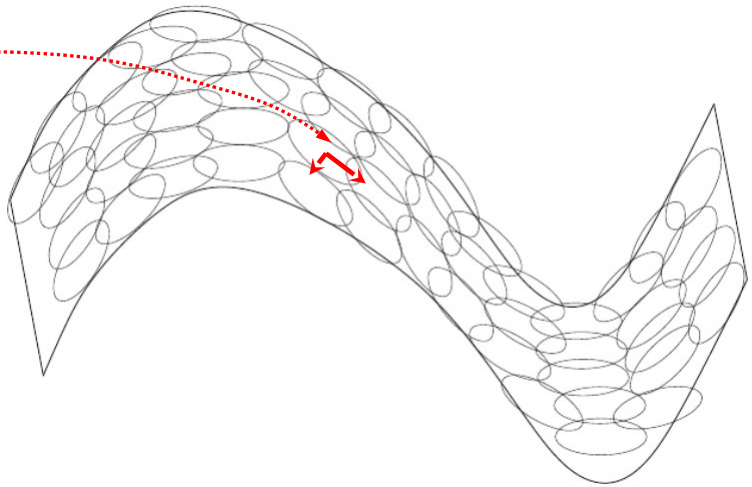


vs.

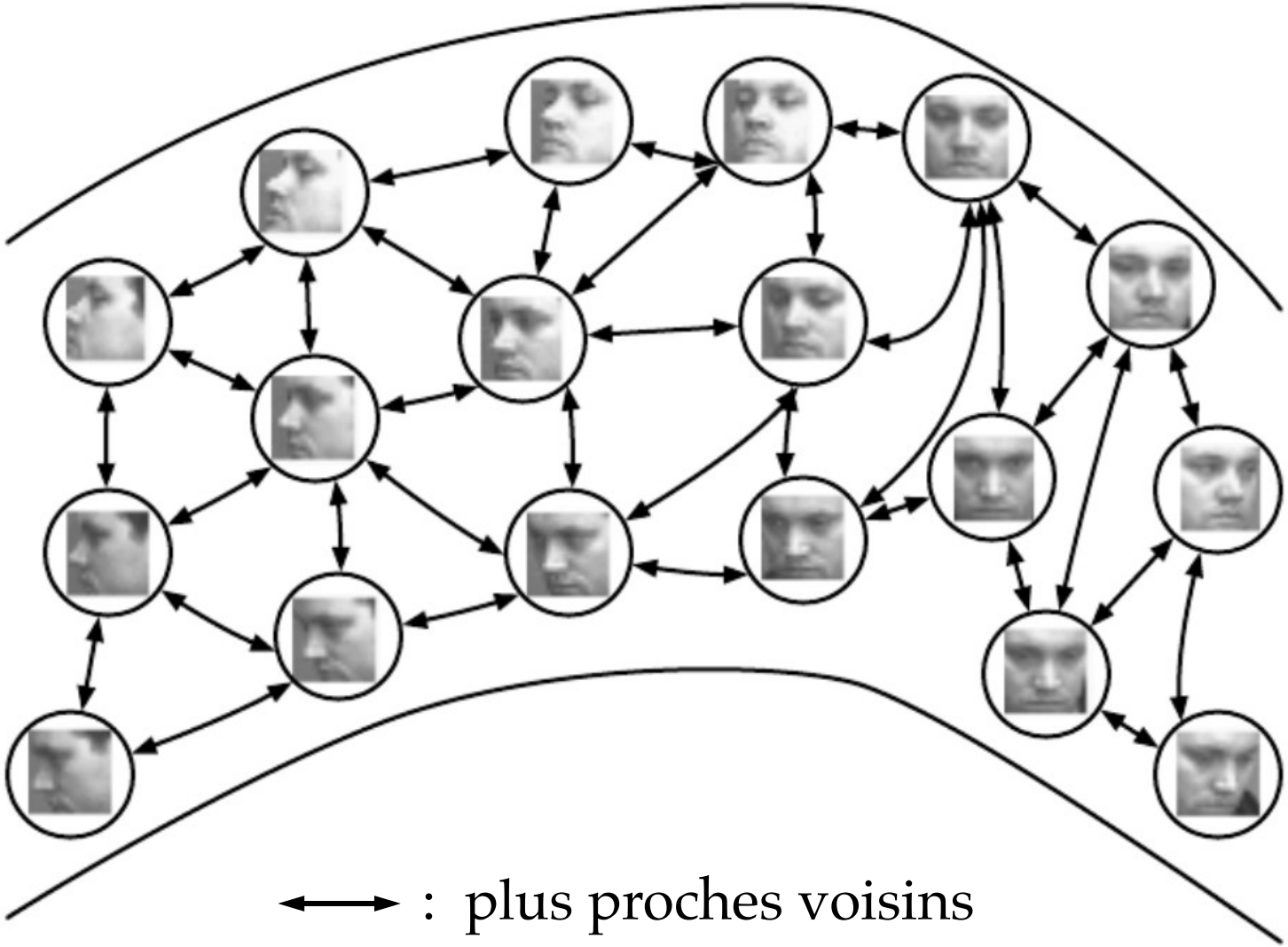
- Compression de x possible car réseau n'a pas à gérer les cas en dehors de la variété

Variété (*manifold*)

- Idéalement, l'encodeur trouvera les variations pertinentes
  - apprendre la « surface » de la variété (tangente)
- Formuler l'architecture pour encourager un type de variété particulier

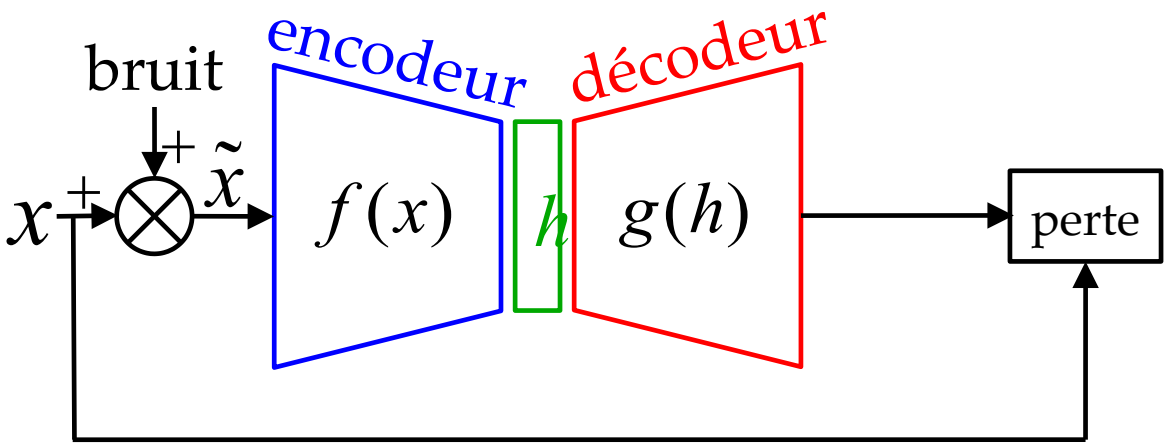


Exemple de variété



Autoencodeur «denoising»

- Ajoute du bruit aléatoire à l'entrée x

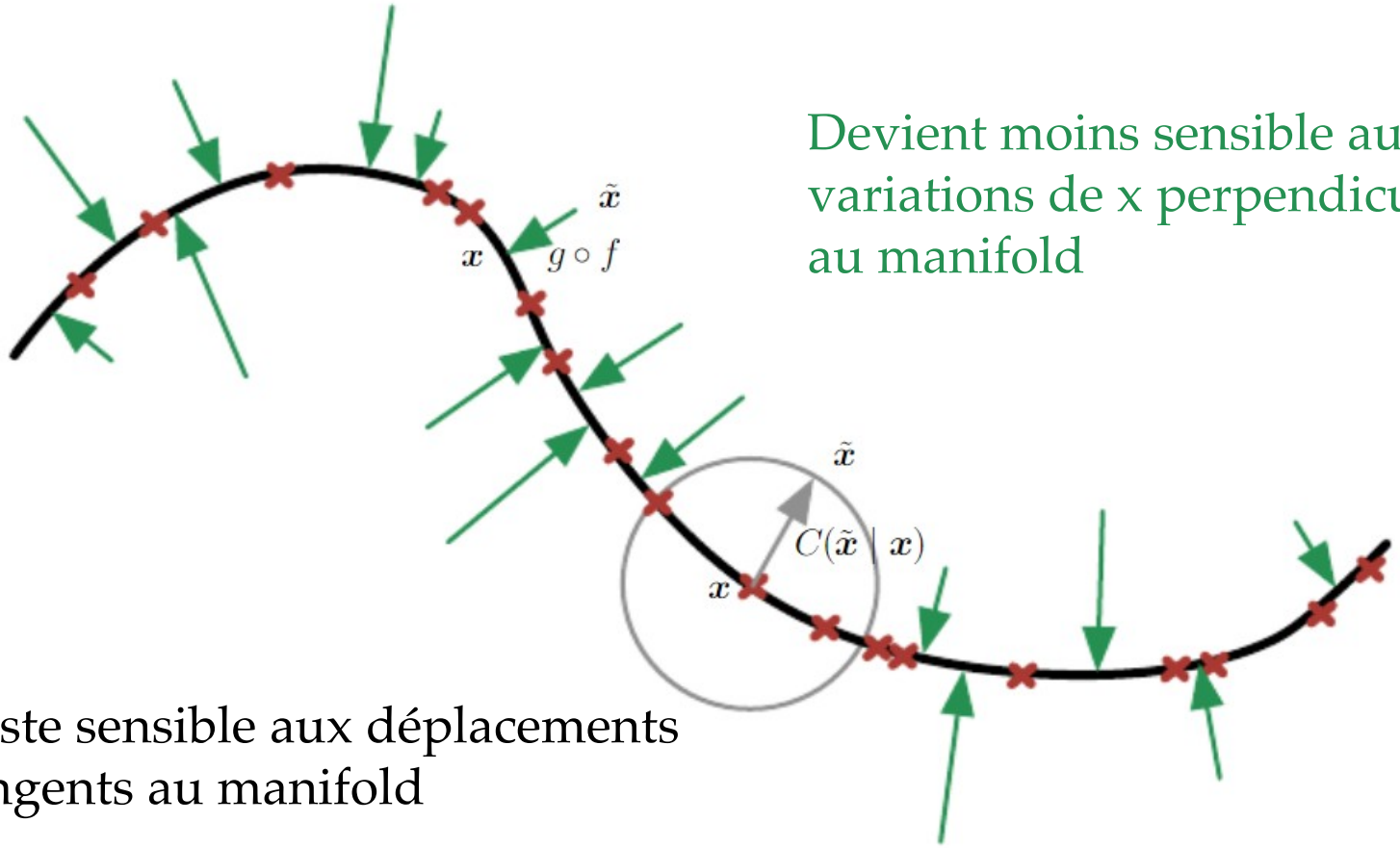


- Cherche quand même à reconstruire x

$$L\left(x, g(f(\tilde{x}))\right)$$

Autoencodeur «denoising»

- Apprend à déplacer des entrées corrompues $\tilde{x}$ vers le manifold

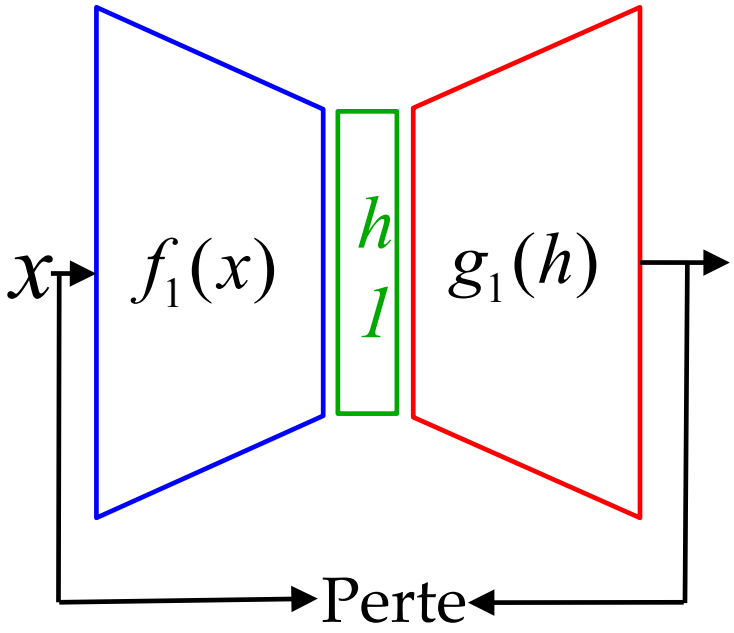


Devient moins sensible aux variations de x perpendiculaires au manifold

Reste sensible aux déplacements tangents au manifold

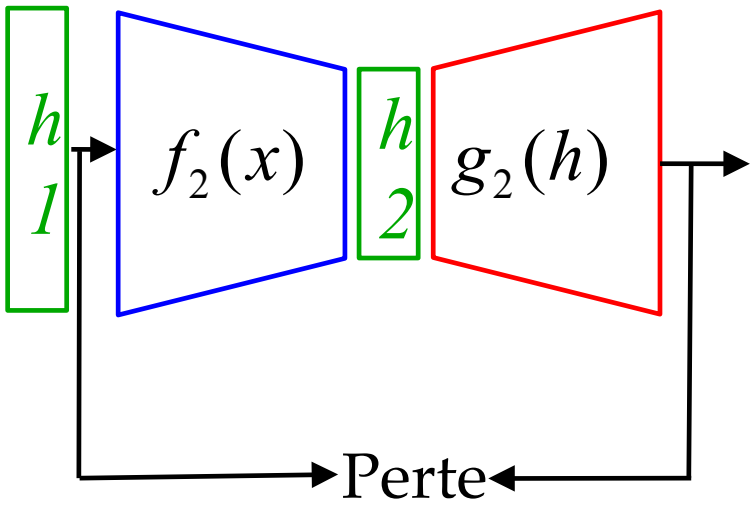
Autoencodeurs profonds

- Si difficulté d'apprendre un autoencodeur profond, on procède par étape (couche par couche)



Autoencodeurs profonds

- Si difficulté d'apprendre un autoencodeur profond, on procède par étape (couche par couche)



Word2vec

Représentation d'un mot par un vecteur

- Soit un dictionnaire de N mots. Chaque mot peut être représenté par un vecteur «one-hot» de dimension N :
  - *Avion* → *(1, 0, 0, …, 0)*
  - *Ballon* → *(0, 1, 0, …, 0)*
  - …
  - *Zèbre* → *(0, 0, 0, …, 1)*

- Dans cet espace, la distance Euclidienne entre deux mots-vecteurs est toujours la même :

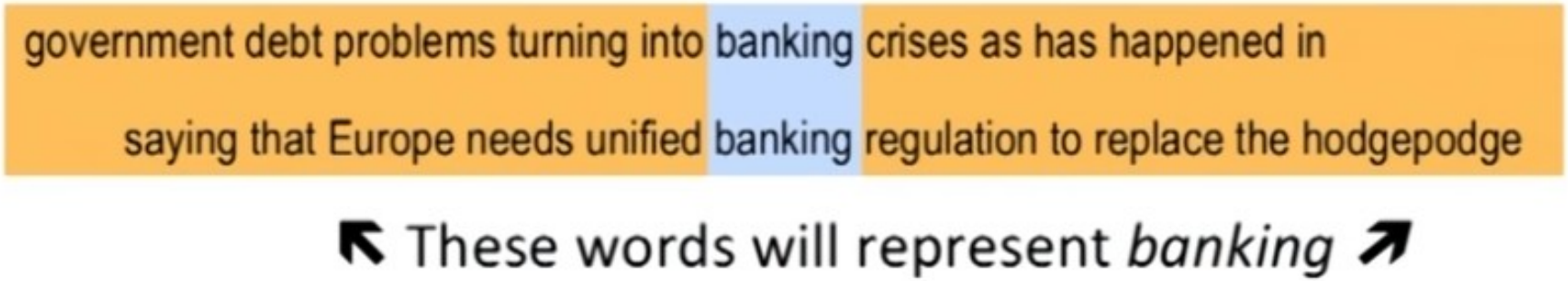
$$\|\text{«avion»} - \text{«ballon»}\| = \|(1,-1,0,\ldots,0)\| = \sqrt{2}$$

Word2vec

- Méthode d'apprentissage non-supervisée de représentations pour les mots
- Chaque mot est représenté par un vecteur dans un espace de valeurs réelles
- Appris sur de larges corpus de textes
- La représentation d'un mot dépends du contexte dans lequel il est utilisé.

Contexte du voisinage

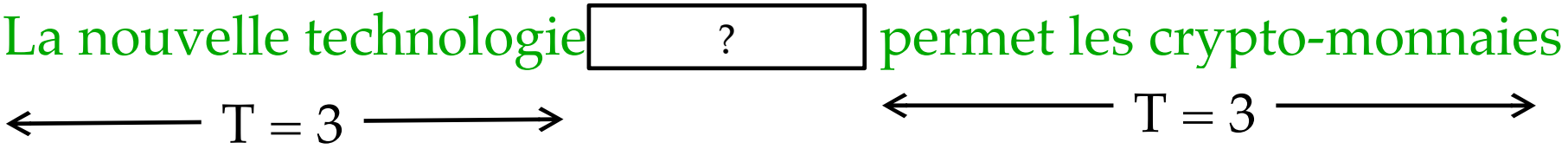
- Vous obtenez beaucoup d'information sur le sens d'un mot en regardant son voisinage dans une phrase



- Chercher un encodage (embedding) qui permet de prédire un/des mots voisins

Entraînement par prédiction

- L'apprentissage des vecteurs se fera via deux tâches
- **Tâche 1** : prédire le mot au centre d'un contexte de ±T

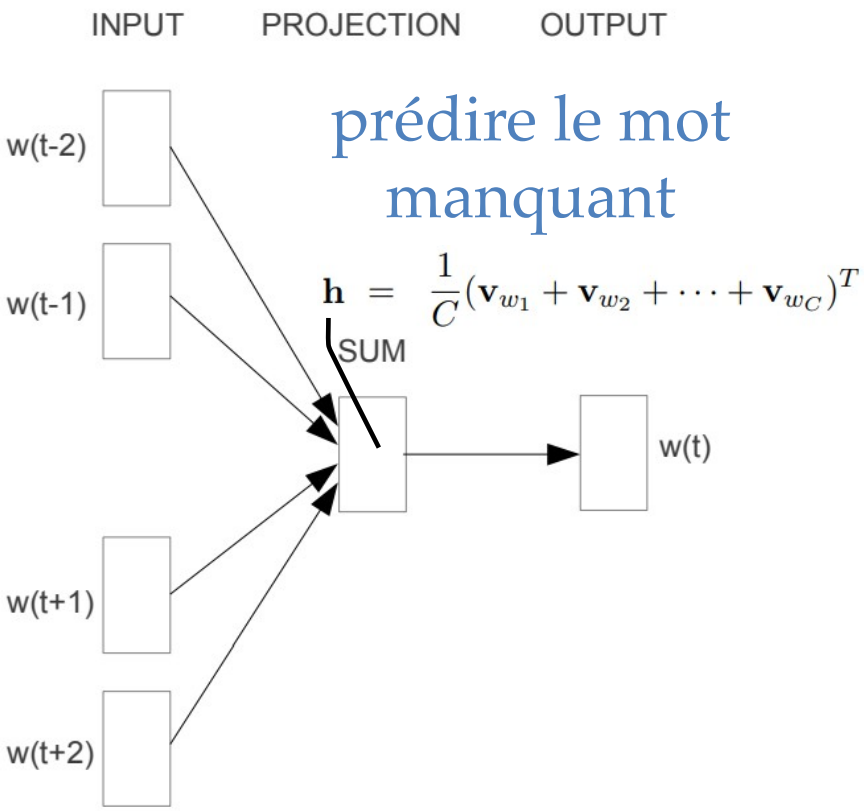


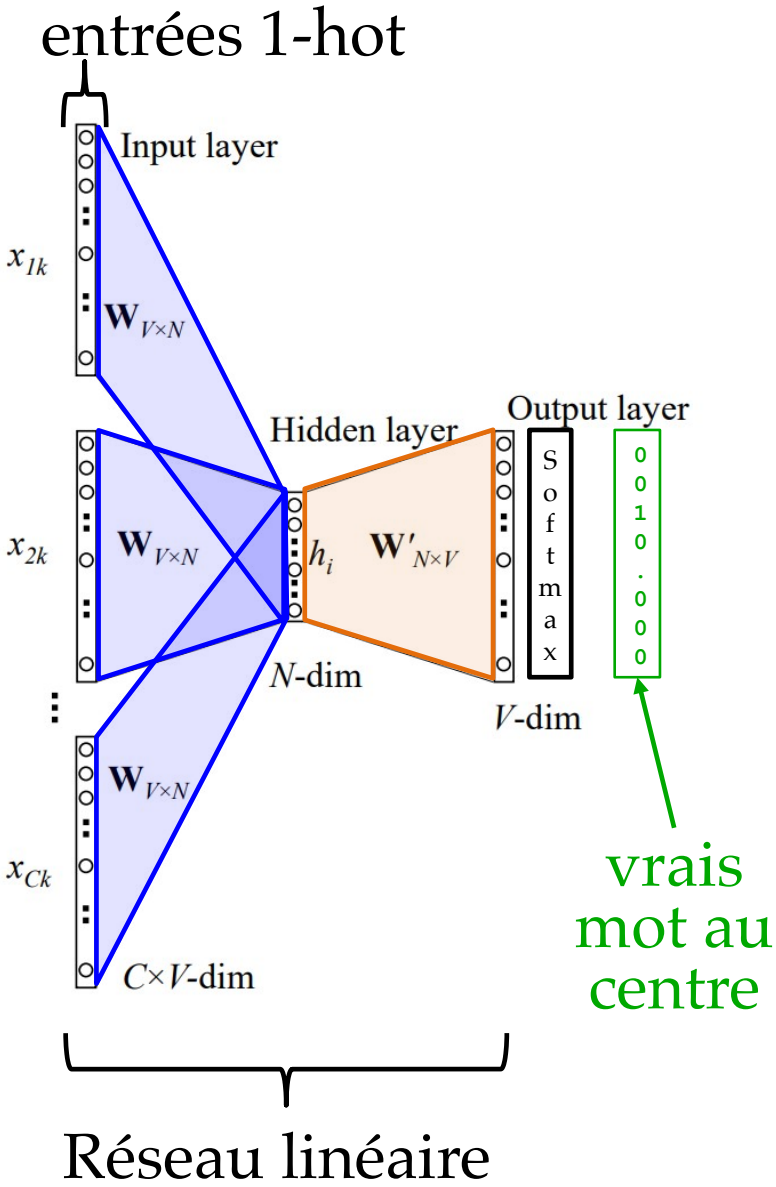
- **Tâche 2** : prédire les mots voisins d'un mot central, pour un contexte de ±T

Tâche 1 : prédiction d'un mot

CBOW (continuous bag-of-words)

(ne tient pas compte de l'ordre des mots)

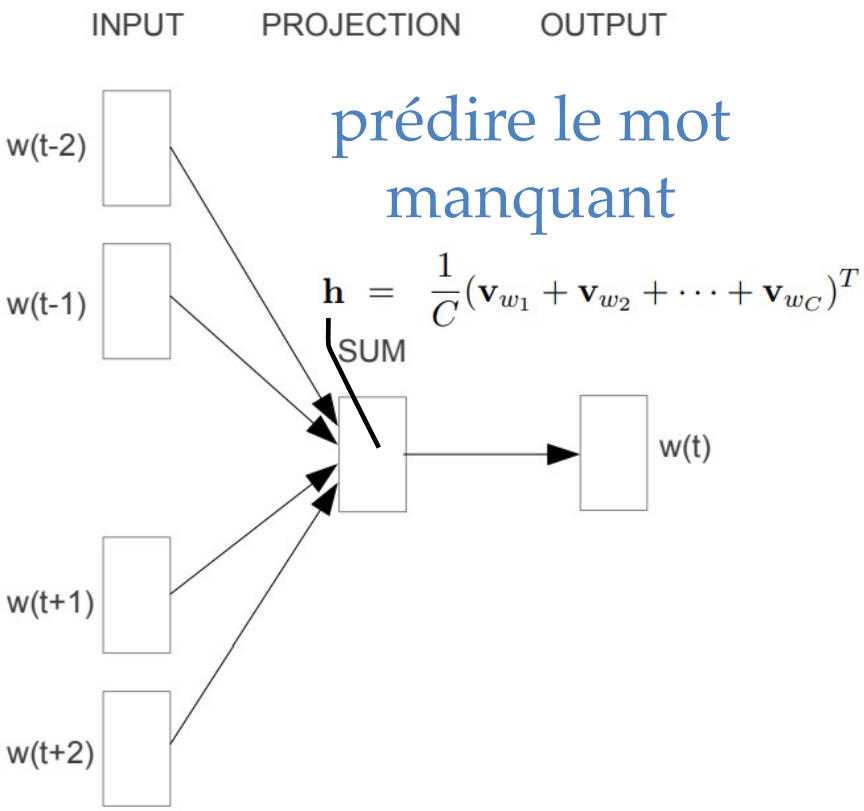


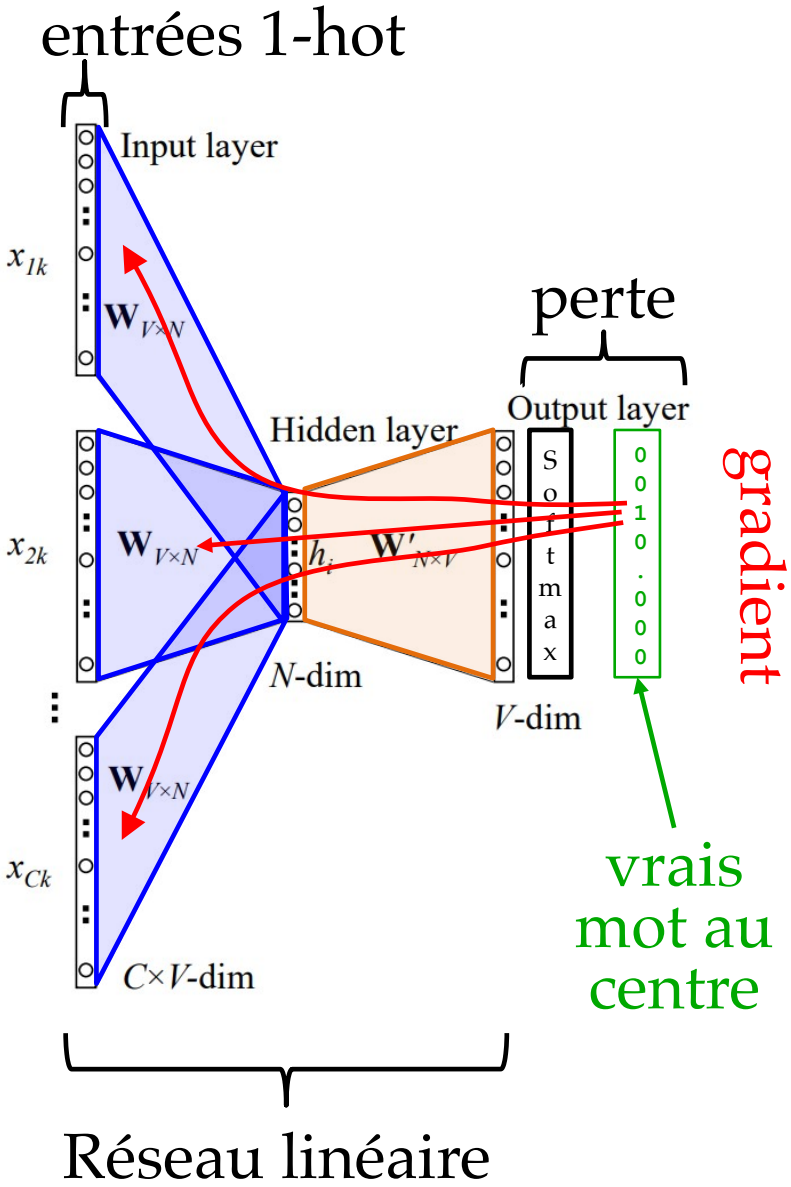


Tâche 1 : prédiction d'un mot

CBOW (continuous bag-of-words)

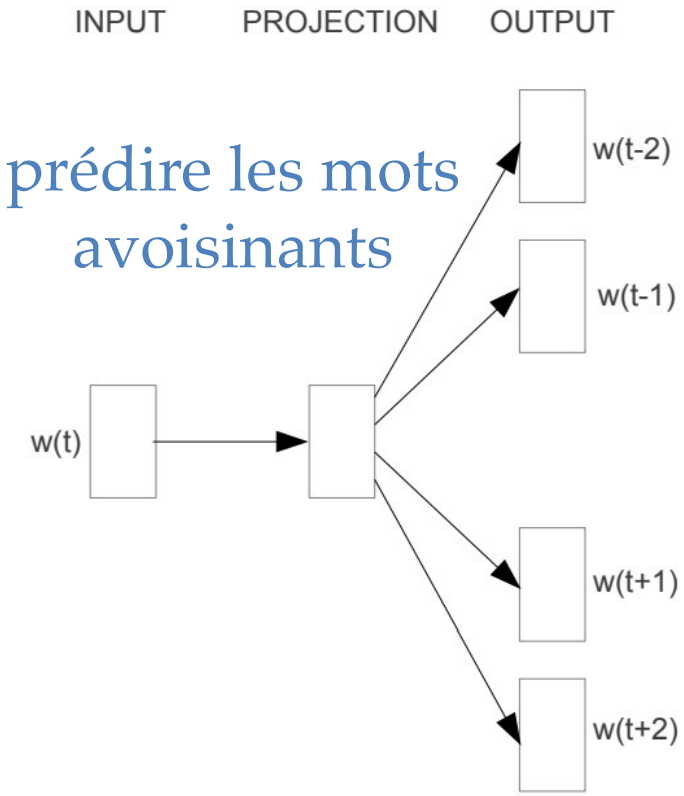
(ne tient pas compte de l'ordre des mots)

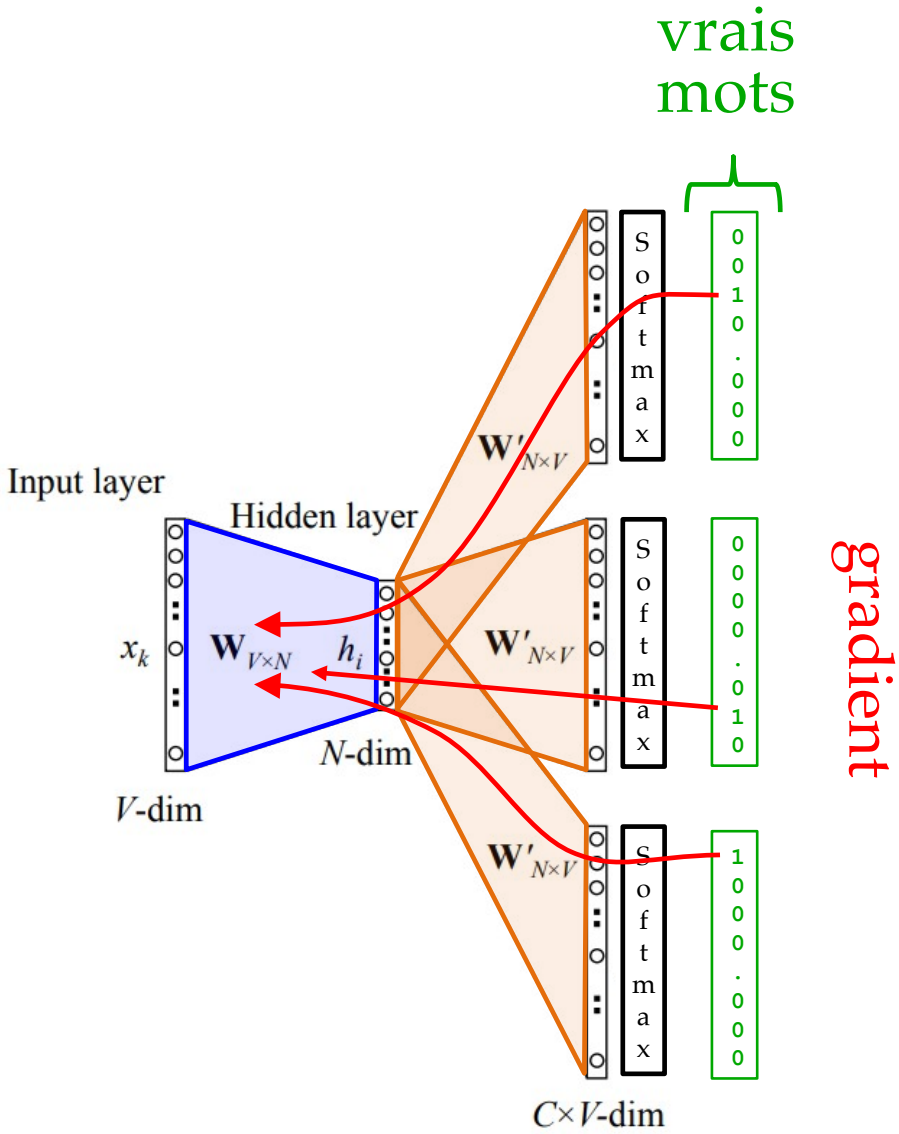




Tâche 2 : prédire mots voisins

Architecture skip-gram





Arithmétique sur embeddings?

- Quelle est la réponse à cette énigme :

France – Paris + Italy = ?

Rome!

Algèbre sur ces vecteurs

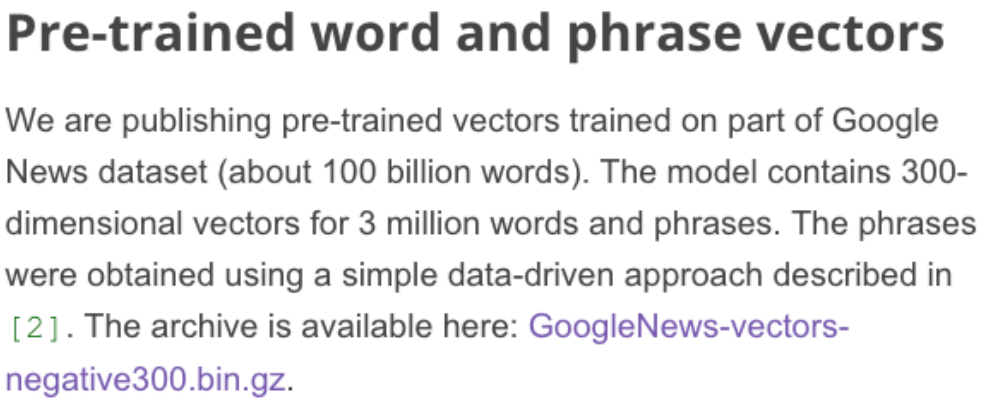
France – Paris + Italy = Rome

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

Quelques liens

- Projet word2vec original, incluant les représentations obtenues par Google

https://code.google.com/archive/p/word2vec/



Pre-trained word and phrase vectors

We are publishing pre-trained vectors trained on part of Google News dataset (about 100 billion words). The model contains 300-dimensional vectors for 3 million words and phrases. The phrases were obtained using a simple data-driven approach described in [2]. The archive is available here: GoogleNews-vectors-negative300.bin.gz.

Quelques liens

- Package Python «Gensim», incluant l'implémentation de word2vec

  (et plusieurs autres algorithmes pour données textuelles)

  https://radimrehurek.com/gensim/models/word2vec.html

- Tutoriel sur l'utilisation de «word embedings» (incluant une discussion intéressante sur les biais sociaux que peuvent contenir les représentations apprises automatiquement)

  https://github.com/fastai/word-embeddings-workshop

Réseau de neurones récurrents

RNN : Recurent Neural Networks

Pourquoi ?

- Traiter des données séquentielles

  Image X vs. $\{x^{(1)}, x^{(2)}, \cdots, x^{(\tau)}\}$

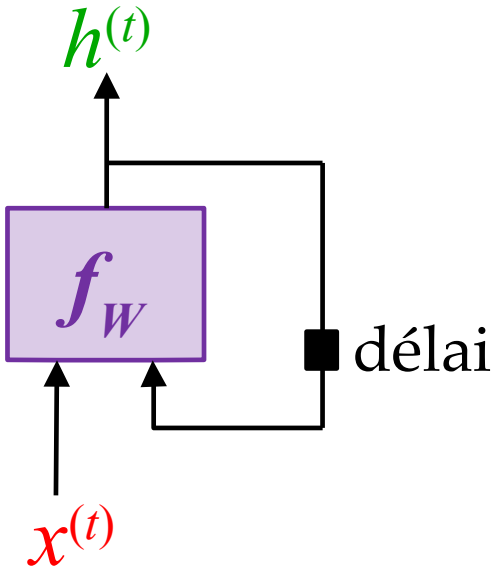
  – séries temporelles
- Souvent de longueur *variable*
- L'information pertinente n'est pas toujours située au même endroit

I went to Nepal in 2009

In 2009, I went to Nepal

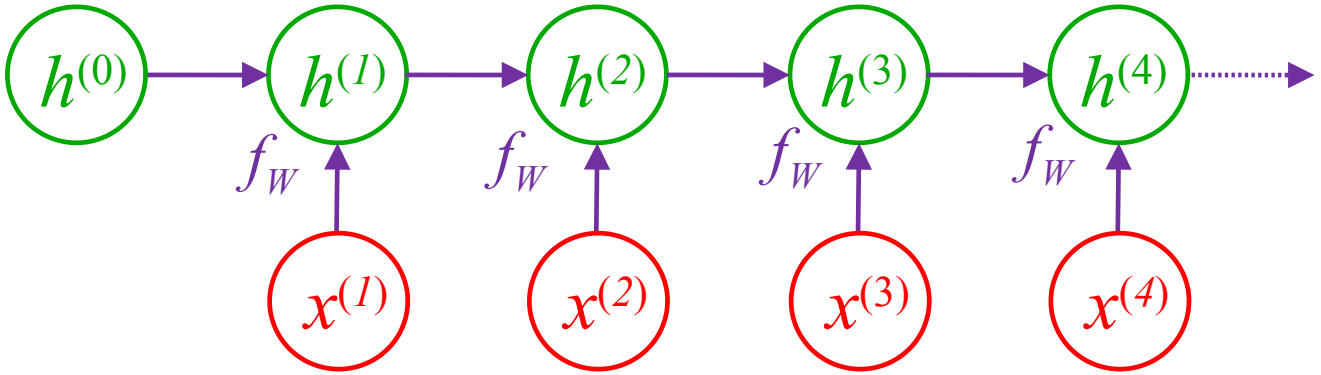
Idée générale

$$h^{(t)} = f_W(h^{(t-1)}, x^{(t)})$$

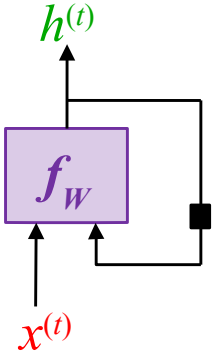


- Relation f_W stationnaire :

  W ne change pas selon t



Variable cachée h



- **Résumé sémantique** de la séquence
- En lien direct avec la tâche :
  - p. e. si on cherche des dates, des mots comme **`vendredi`** vont influencer h plus que **`Lille`**

La rétropropagation des gradients fera le travail de trouver la fonction f_W favorisant cette représentation

- La taille de h influencera la quantité d'information pouvant y être emmagasiné
  - pourra difficilement résumer *À la recherche du temps perdu* de M. Proust (4 215 pages)

Topologies RNN

one to many

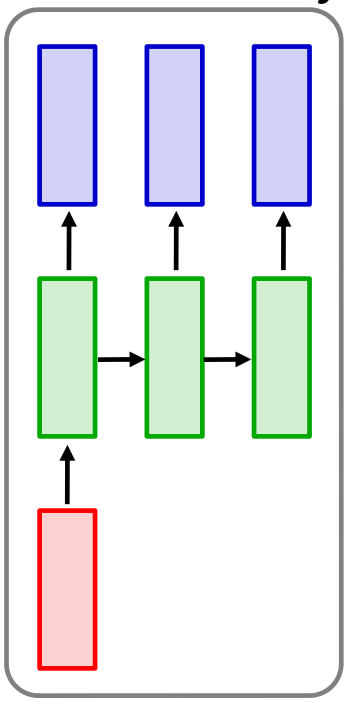
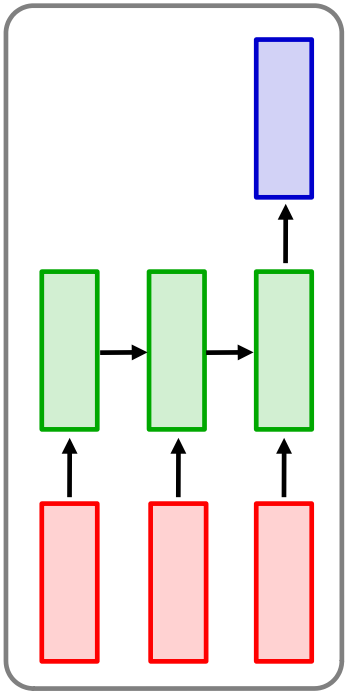


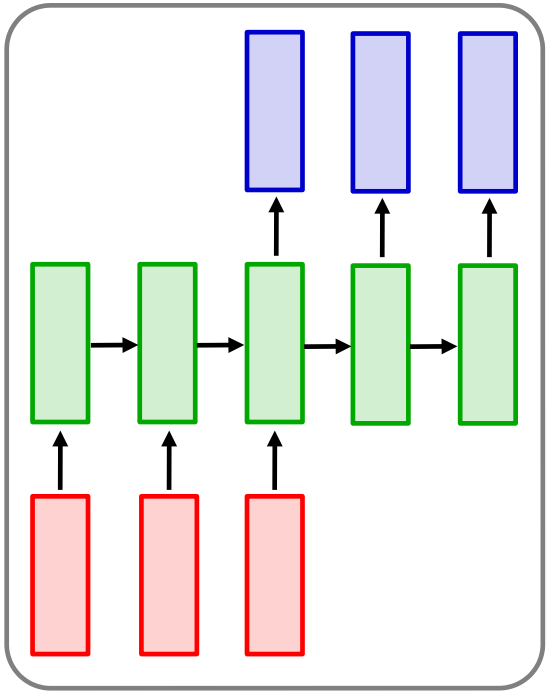
Image captioning

many to one



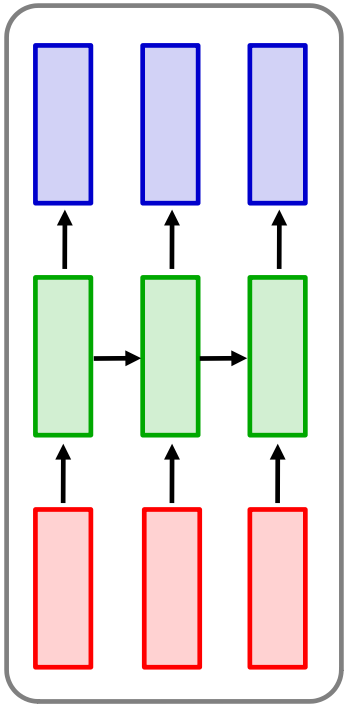
Classification de sentiment (texte)

many to many



Traduction, Réponse aux questions

many to many

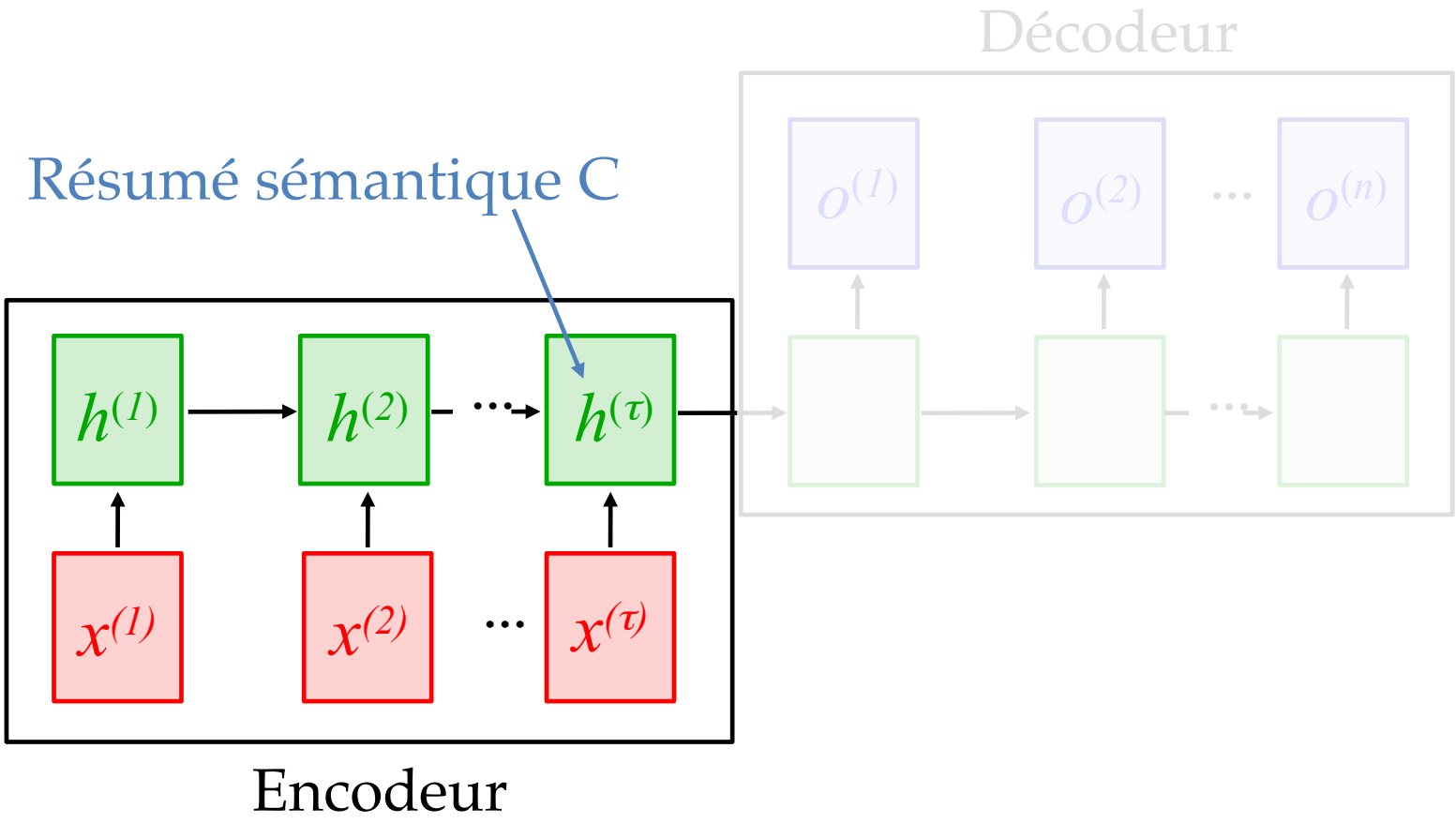


Classification de trames vidéos

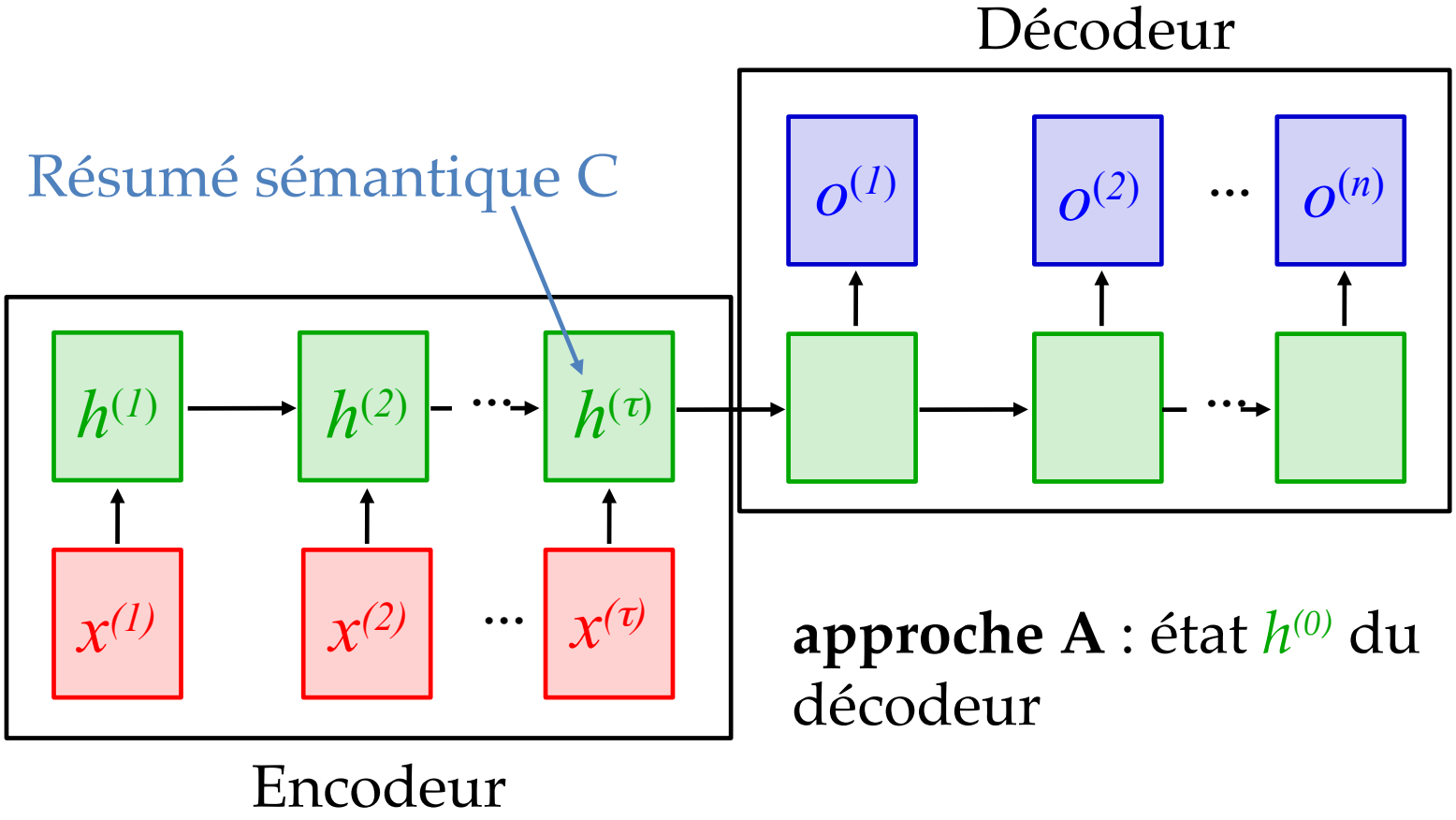
Sequence-to-sequence

- Architecture *many to many*
- Généré une séquence à partir d'un résumé C

contexte

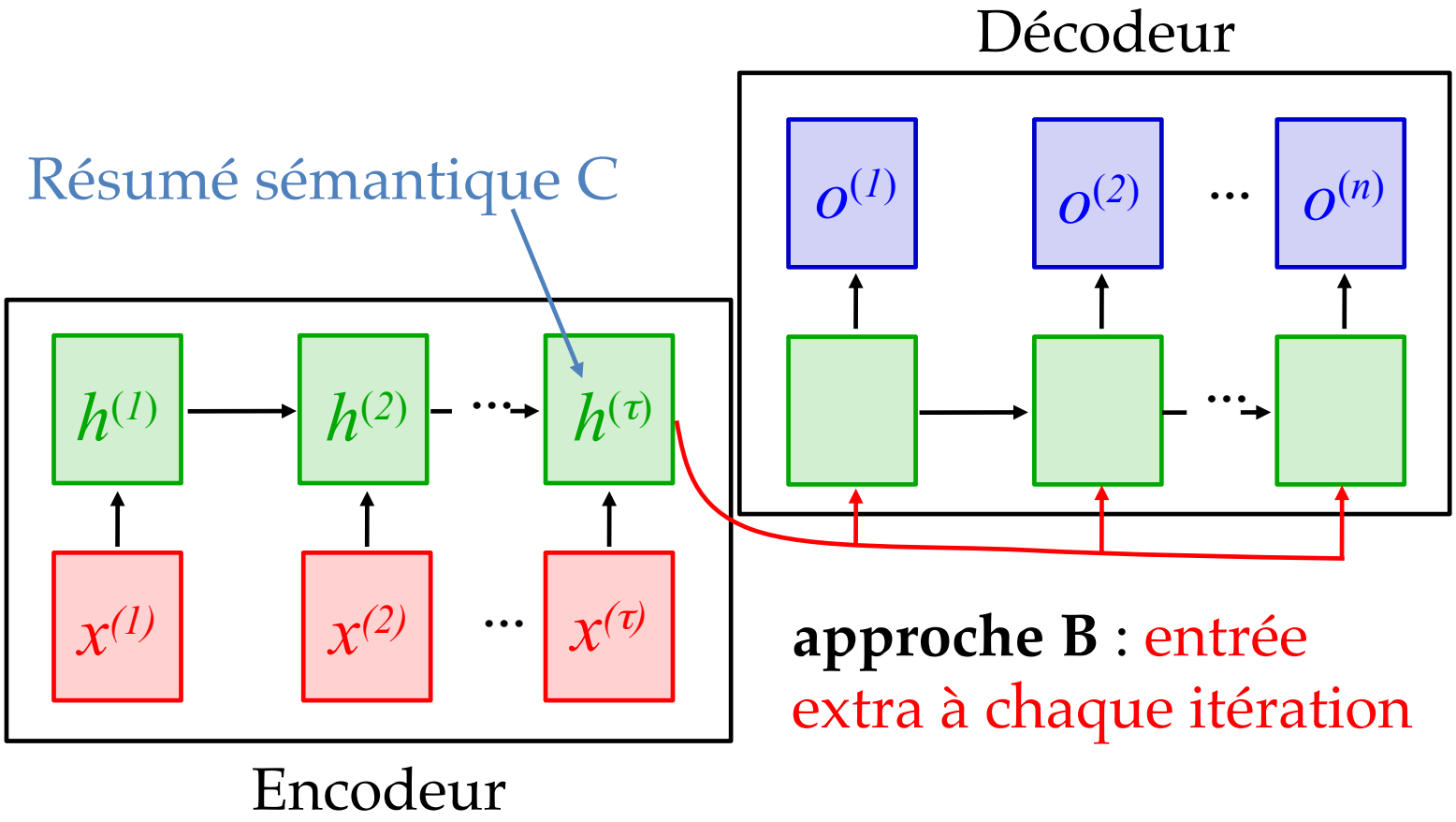


Sequence-to-sequence



approche A : état $h^{(0)}$ du décodeur

Sequence-to-sequence



approche B : entrée extra à chaque itération

Sequence-to-sequence