UNIVERSITÉ LAVAL

GLO-4030/7030
APPRENTISSAGE PAR RÉSEAUX DE NEURONES PROFONDS

Réseaux Récurrents (RNN)

Hiver 2020

Philippe Giguère

Pourquoi RNN ?

- Traiter des données séquentielles

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

  – séries temporelles
  – séquences de pixels
  – séquences de mots
- Souvent de longueur ***variable***
- Pas clair d'avance où l'information pertinente est située

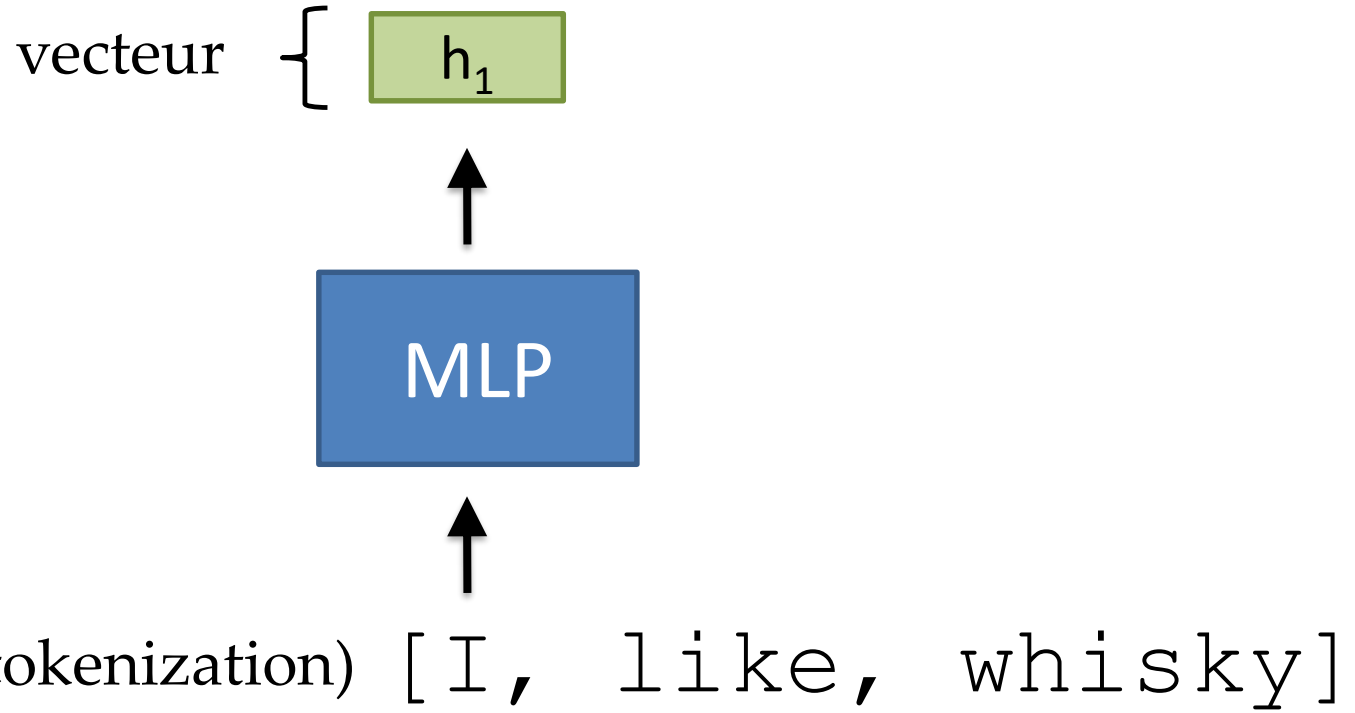
I went to Nepal in 2009

In 2009, I went to Nepal

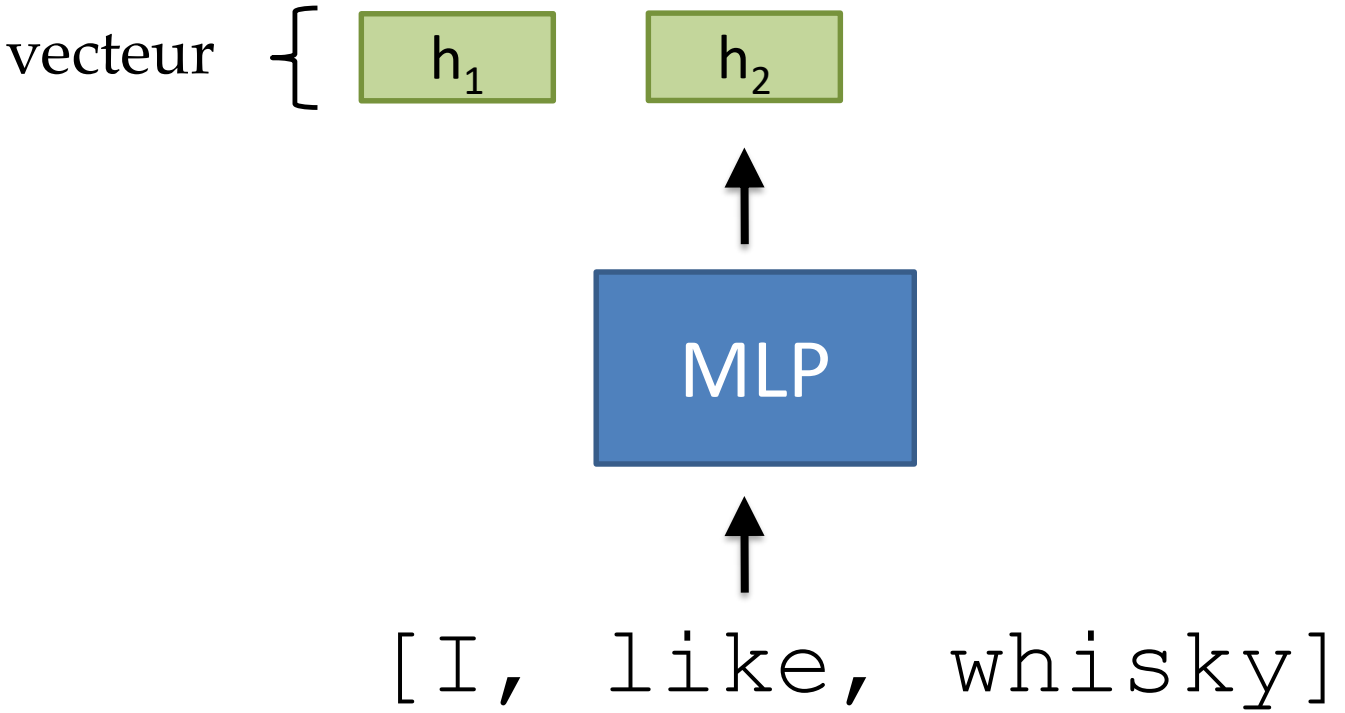
Exemple textuel

```
I like whisky
```

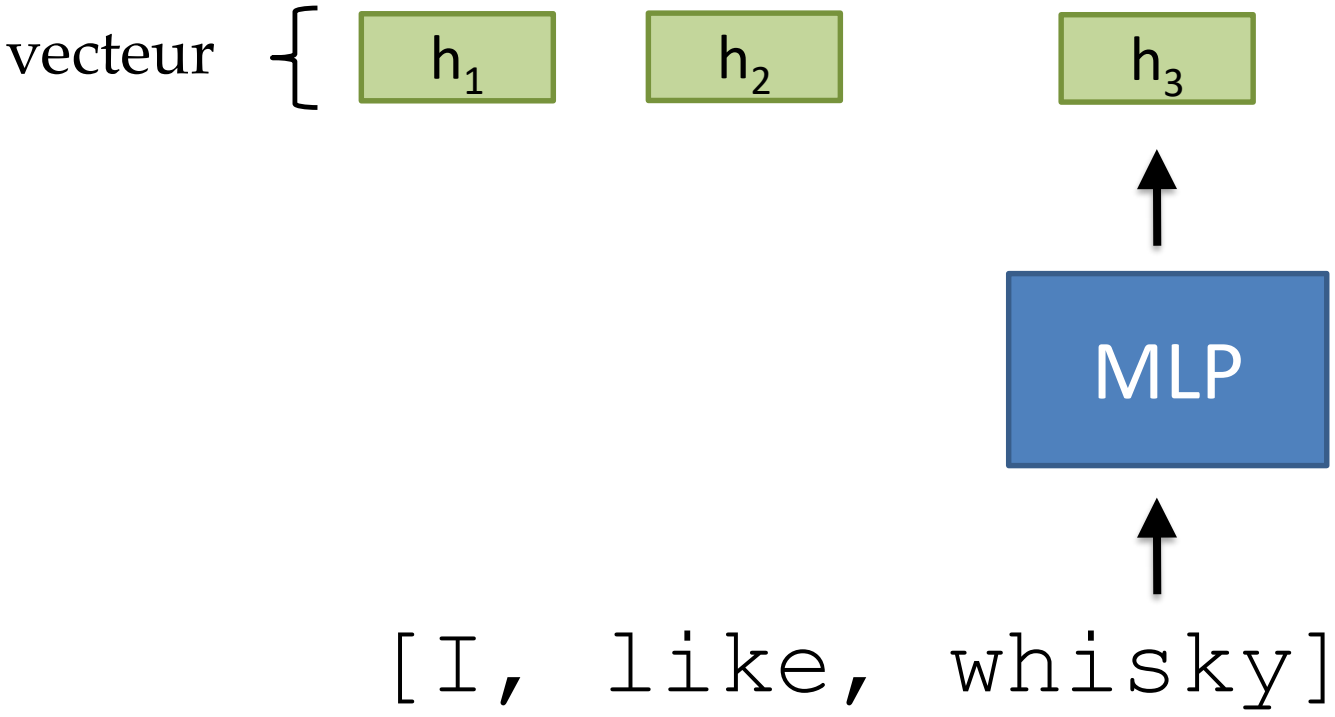
Exemple textuel



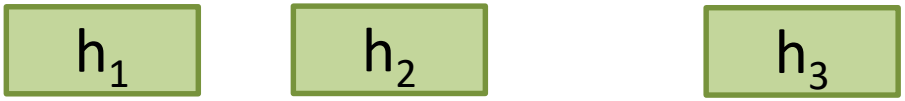
Exemple textuel



Exemple textuel



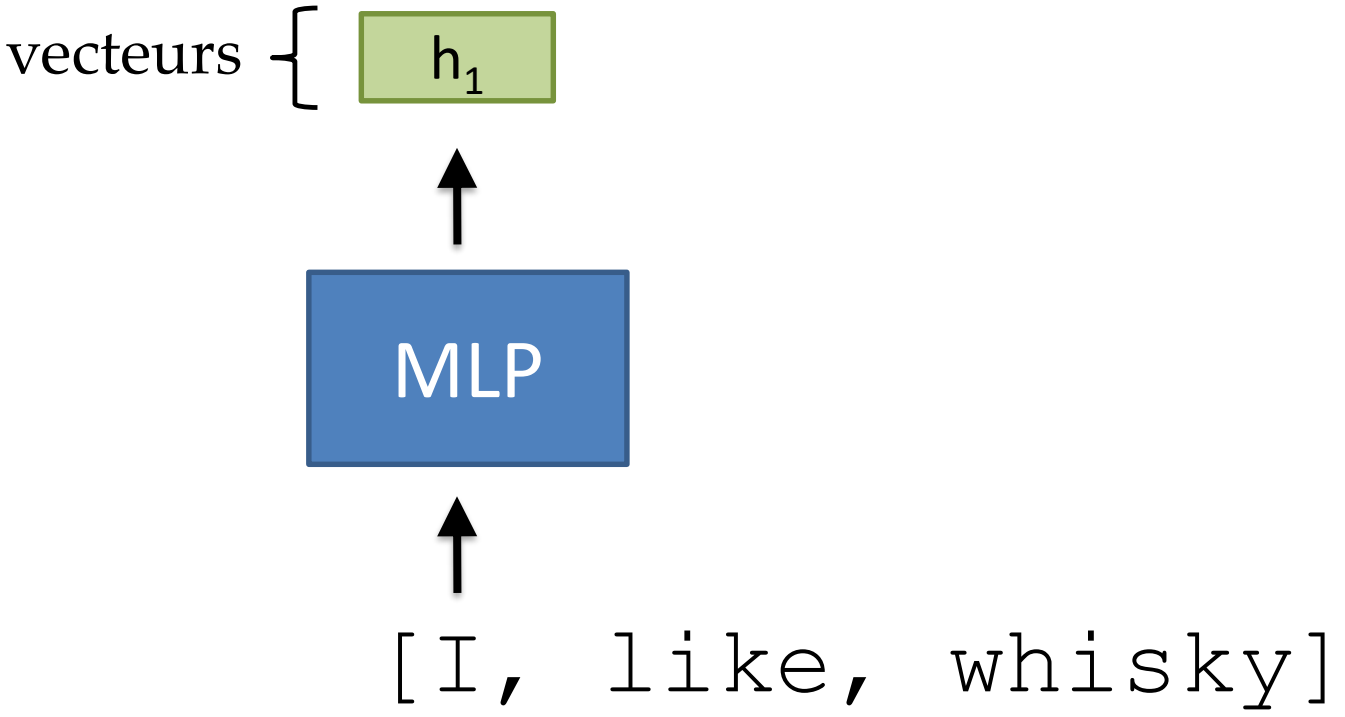
Exemple textuel



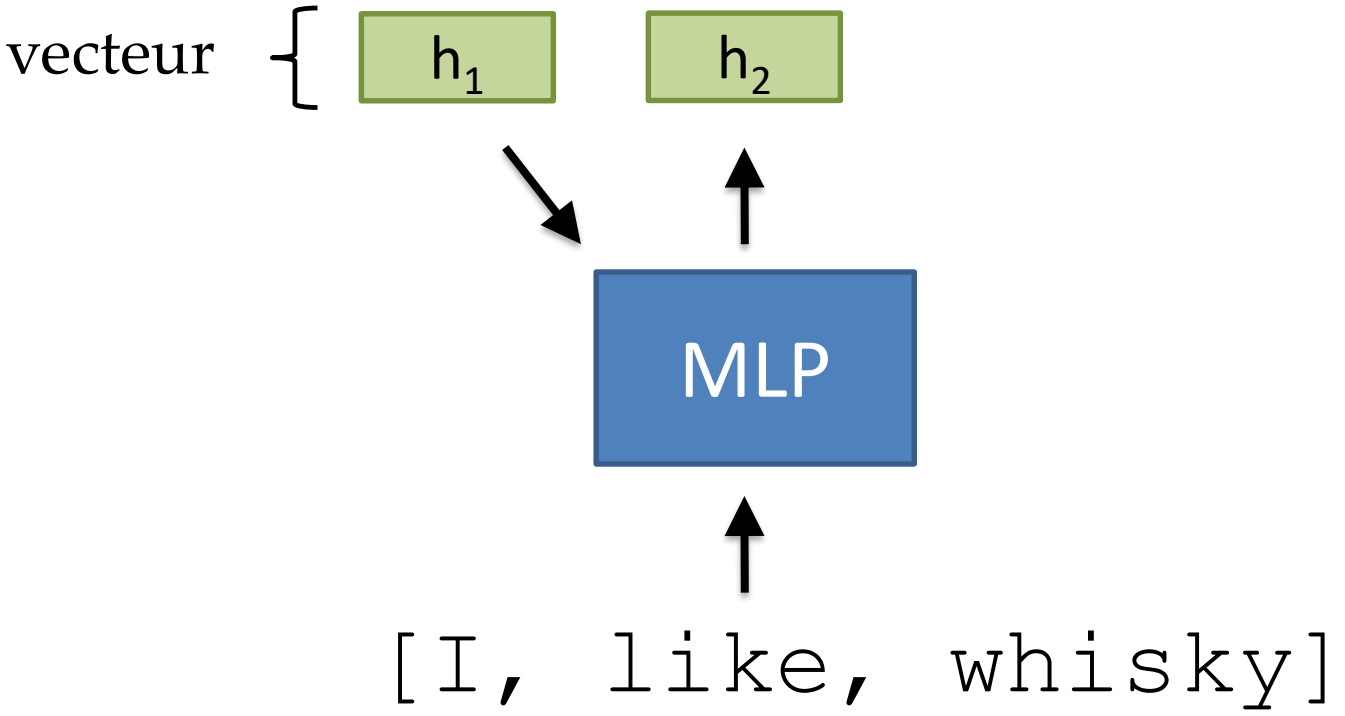
Pas de relation entre les h

```
[I, like, whisky]
```

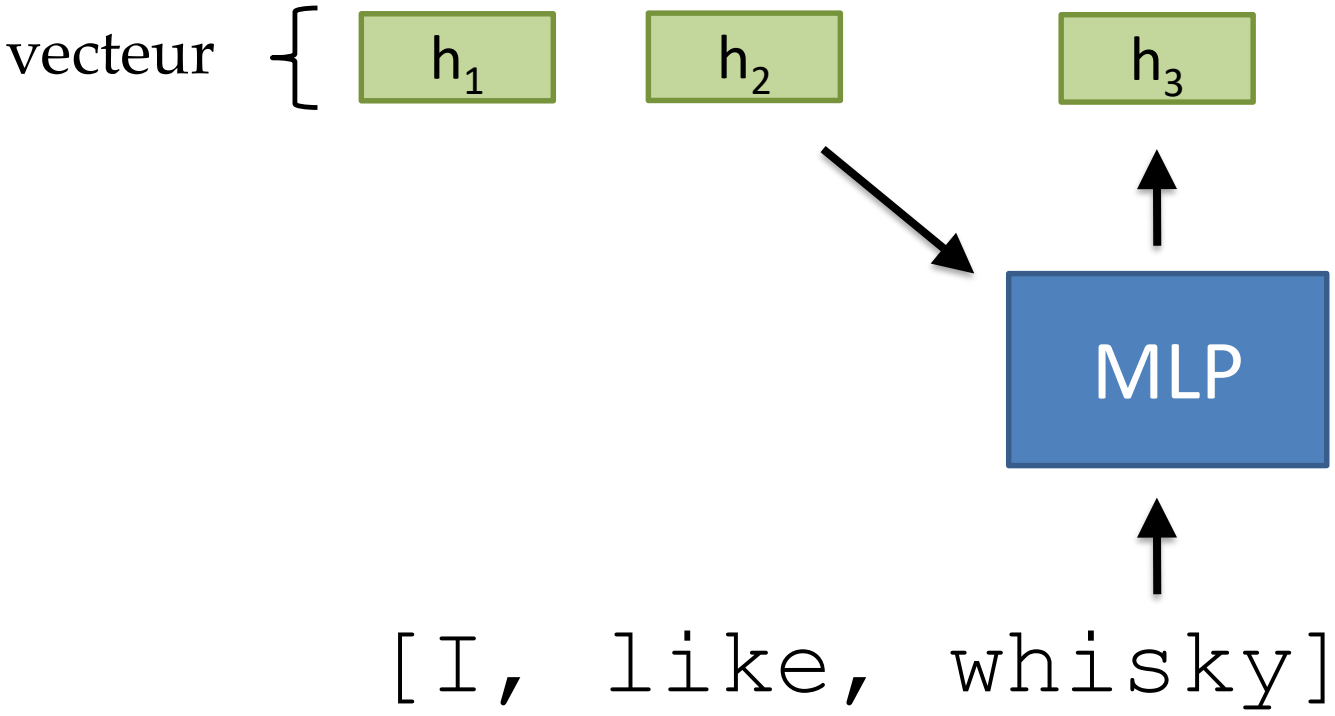
Vers le réseau récurrent …



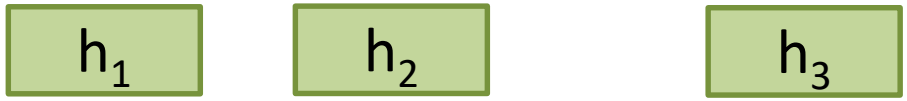
Vers le réseau récurrent …



Vers le réseau récurrent …



Vers le réseau récurrent …



Maintenant h_3 pourra contenir de l'information **sur toute la séquence**

`[I, like, whisky]`

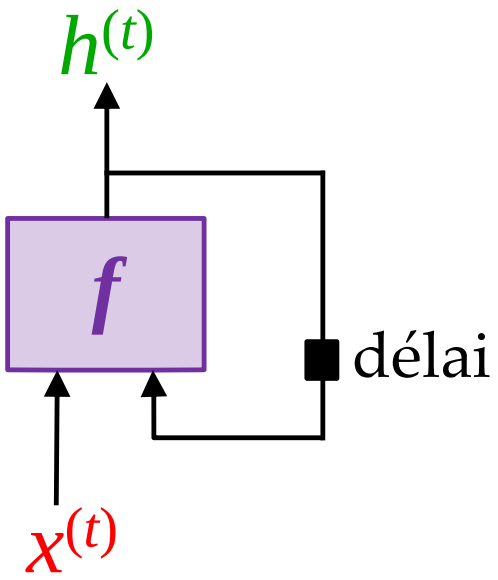
Modélisation séquentielle idéale

1. Être capable de traiter les séquences de longueur variable
2. Garder la trace des dépendances à long-terme
3. Conserver l'information sur l'ordre
4. Partager les paramètres le long de la séquence

Adapté de MIT 6.S191

Idée générale

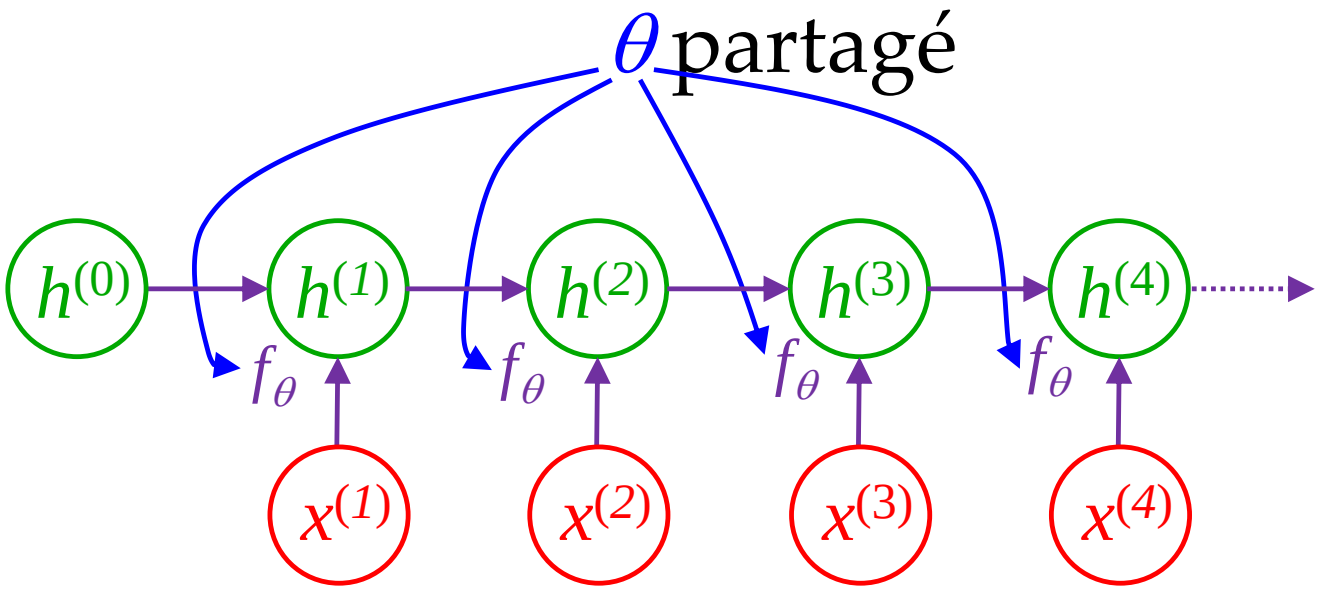
$$h^{(t)}=f(h^{(t-1)}, x^{(t)}, \theta)$$



- Mêmes paramètres θ (*weight sharing*)
- Limite le pouvoir de représentation
  - régularisation
- Relation f stationnaire : ne change pas selon t
  - p. e. règle grammaire indépendante de la position
- Lien avec systèmes dynamiques (GMC, GEL)

Graphe calcul déroulé

$$h^{(t)}=f(h^{(t-1)}, x^{(t)}, \theta)$$



Variable cachée *h*

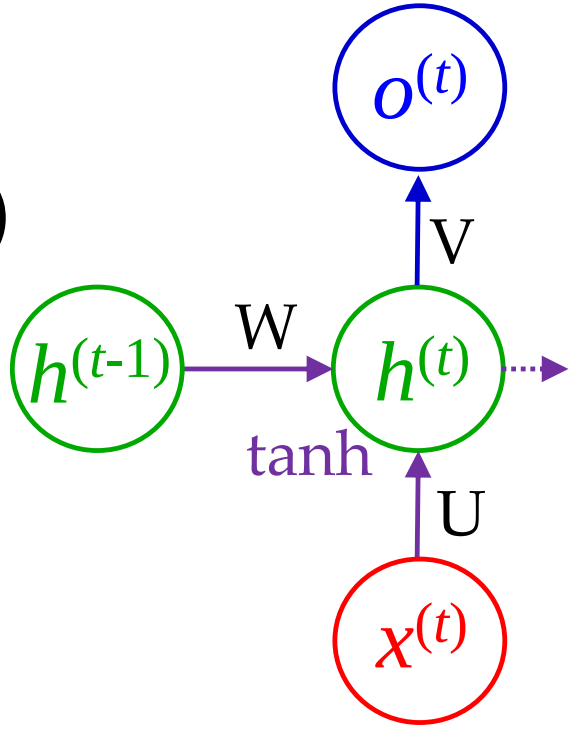
- **Résumé sémantique** (avec perte) de la séquence (passée, si causal)
- En lien direct avec la tâche :
  - p. e. si on cherche des dates, des mots comme **`mercredi`** vont influencer *h* plus que **`Québec`**
  - backprop fera le travail de trouver la fonction *f* favorisant cette représentation
- Taille de *h* influencera la quantité d'information pouvant y être stockée
  - pourra difficilement résumer *À la recherche du temps perdu* de M. Proust (4 215 pages)
  - généralisation plus difficile si *h* est grand

RNN universel (vanille)

- Utilise des fonctions affines
- tanh comme non-linéarité

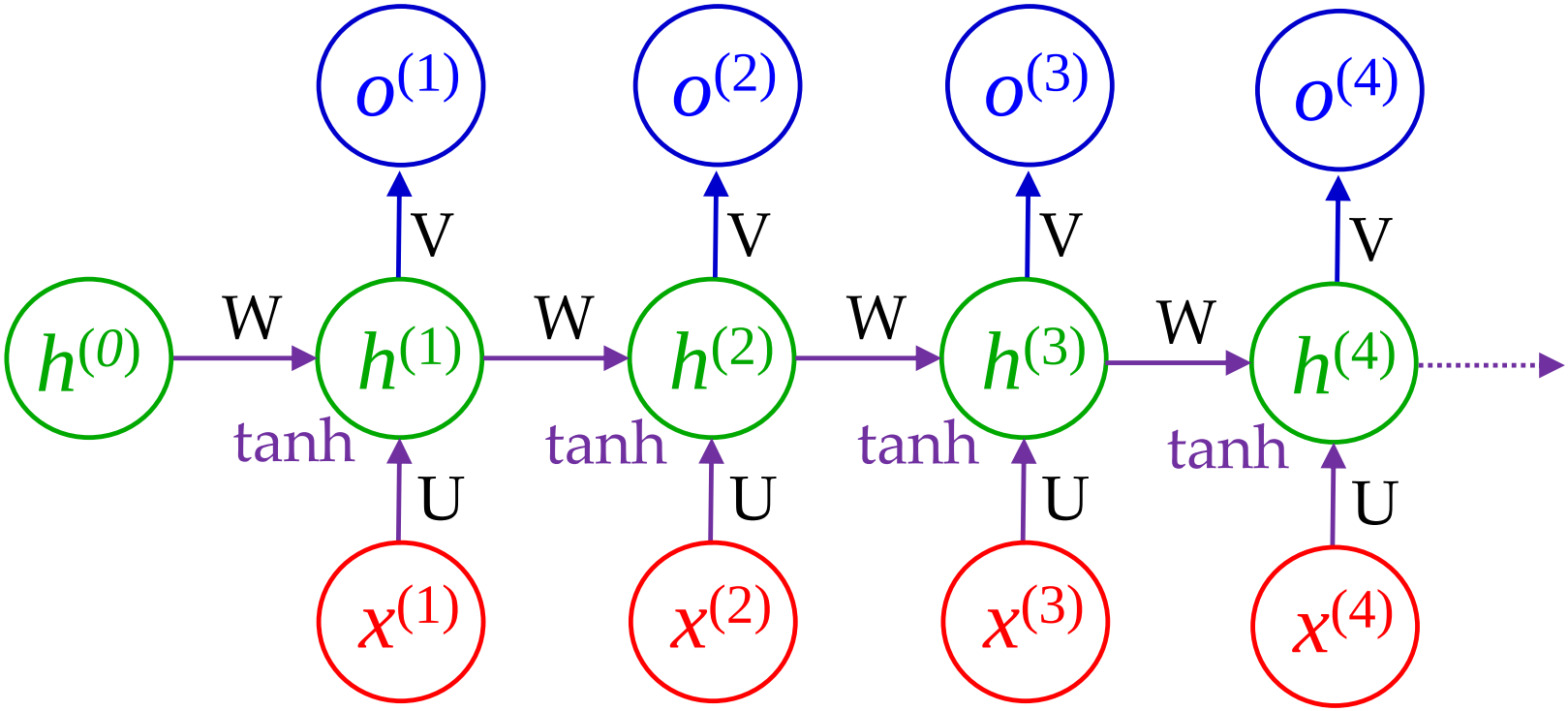
$$h^{(t)} = \tanh(Wh^{(t-1)} + Ux^{(t)} + b)$$

$$o^{(t)} = Vh^{(t)} + c$$



- Peut accomplir autant qu'une machine de Turing
- Variante assez commune
- Défaut : on ne peut pas paralléliser forward/backward pass
  - doit faire la séquence au complet en sériel

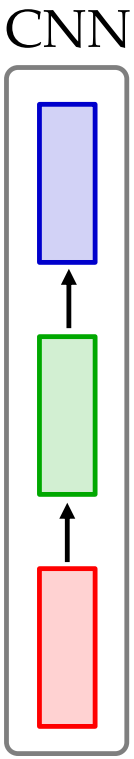
RNN vanille déroulé



Pourquoi tanh ?

- Non-linéaire
- Toujours dérivable
- Sortie [-1,1] (enlever/ajouter)
- Symétrique
- Pas de biais systématique
  - sigmoïde va de [0,1], induit biais
- Autres ?

Topologie Feedforward



Topologie RNN

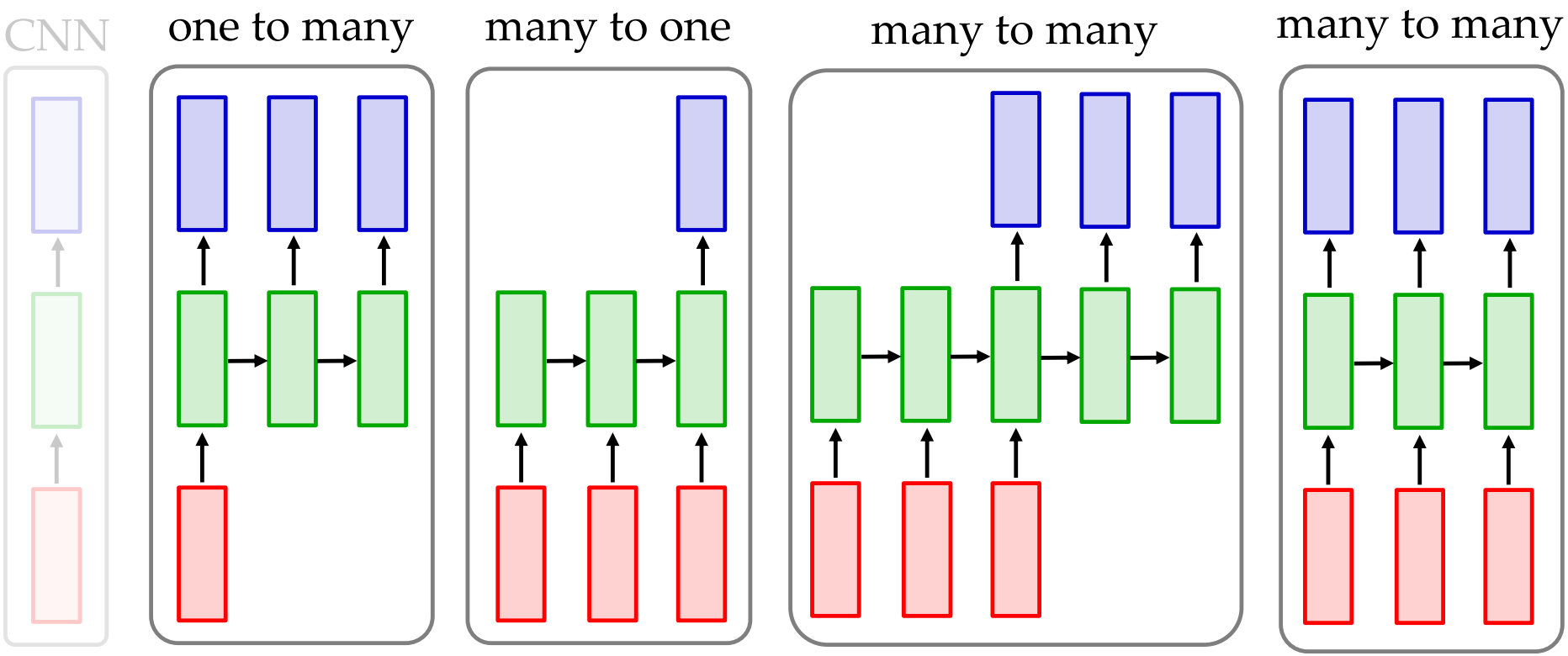


Image captioning

Classification de sentiment (texte)

Traduction, Réponse aux questions (tailles entrée/sortie variables)

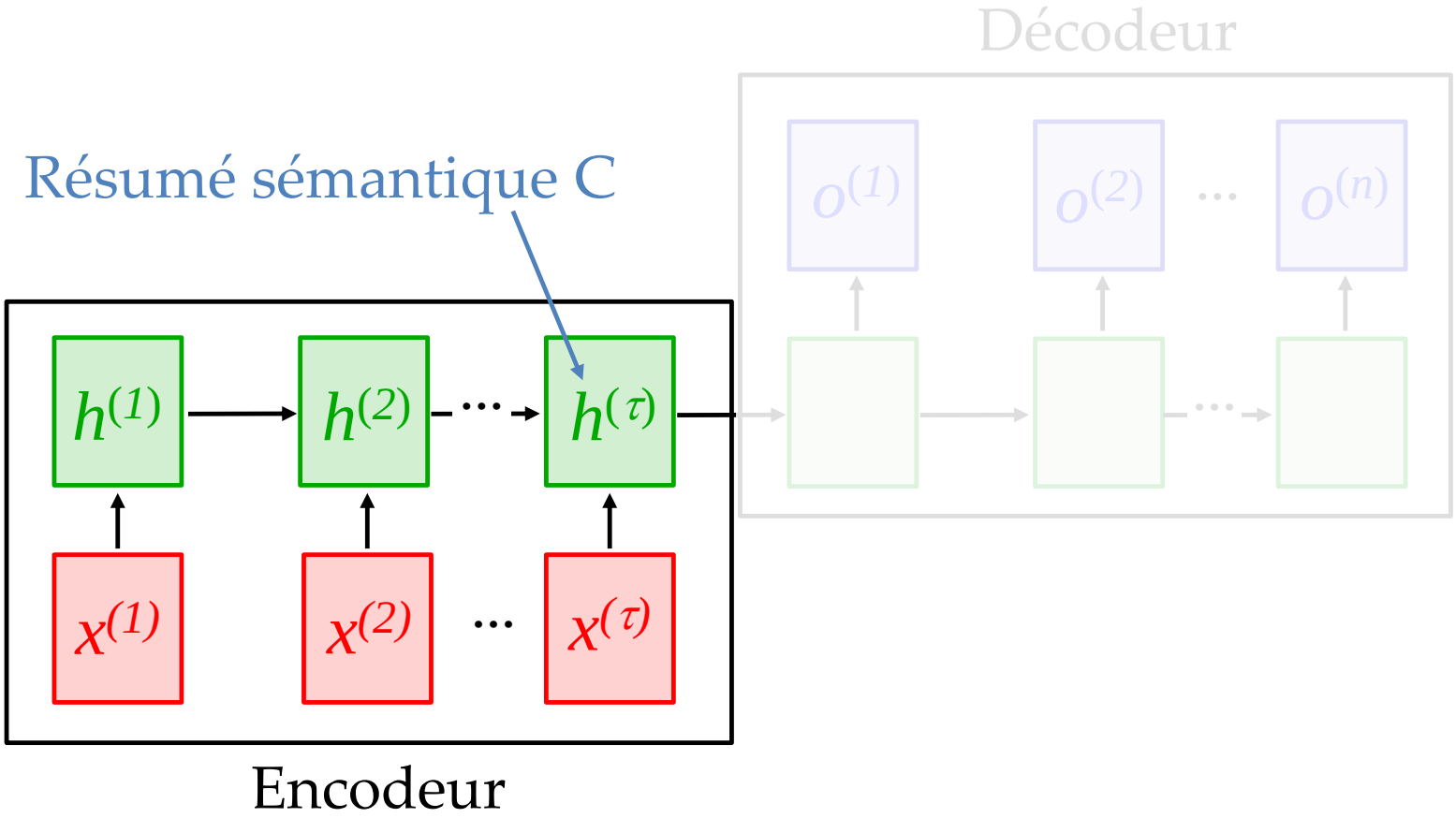
Classification de trames vidéos

adapté de cs231n

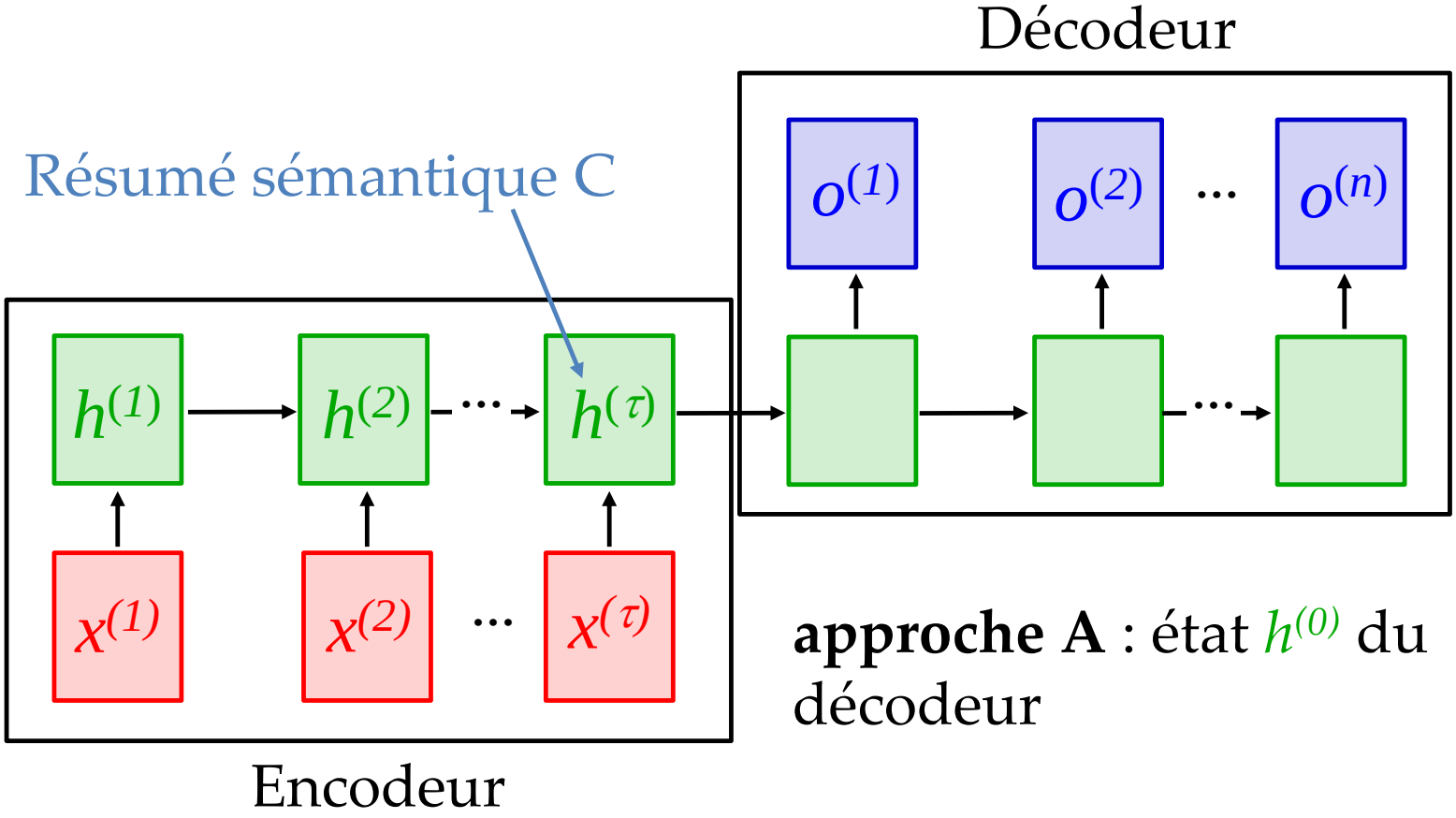
Sequence-to-sequence

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

contexte

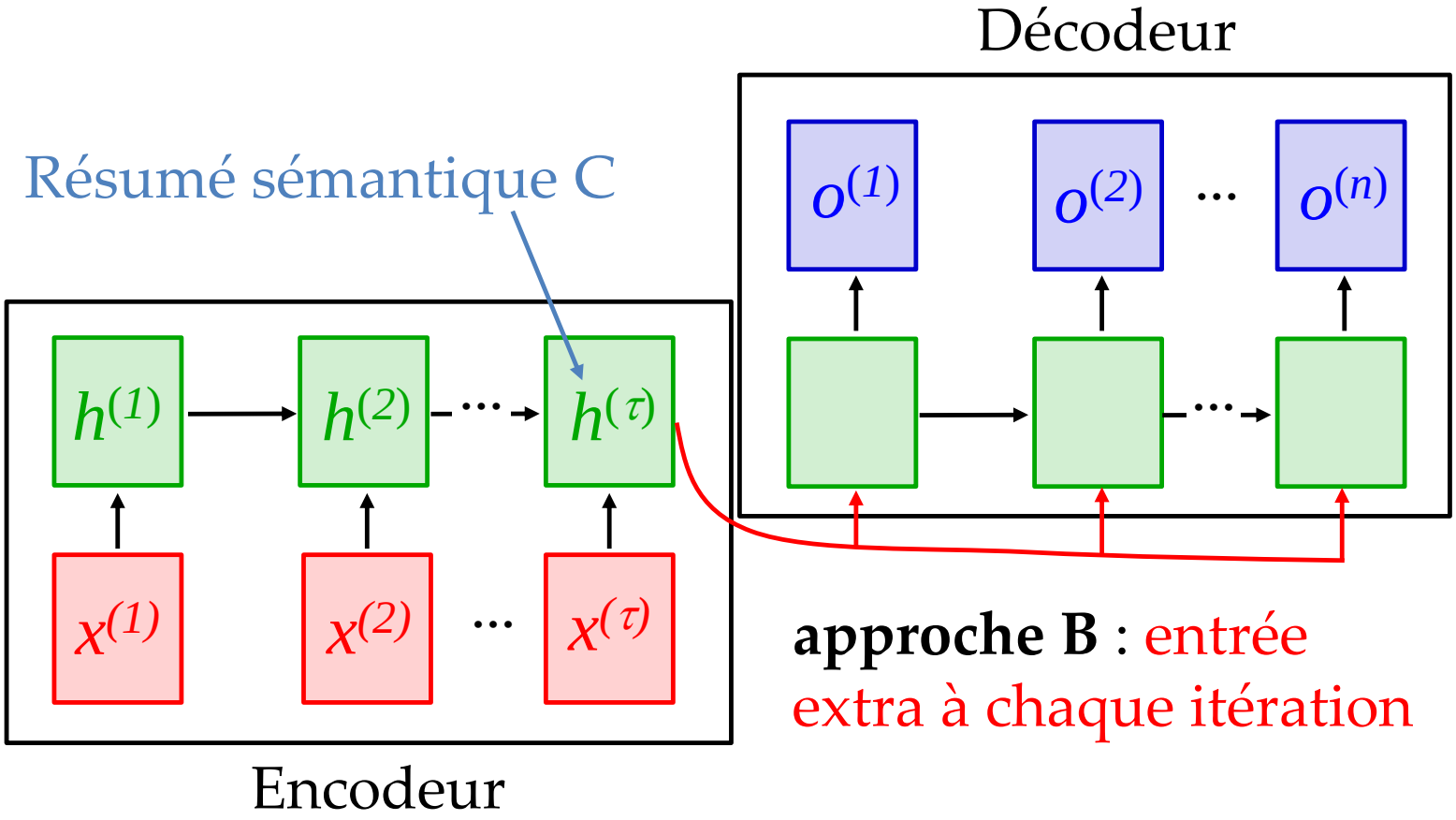


Sequence-to-sequence



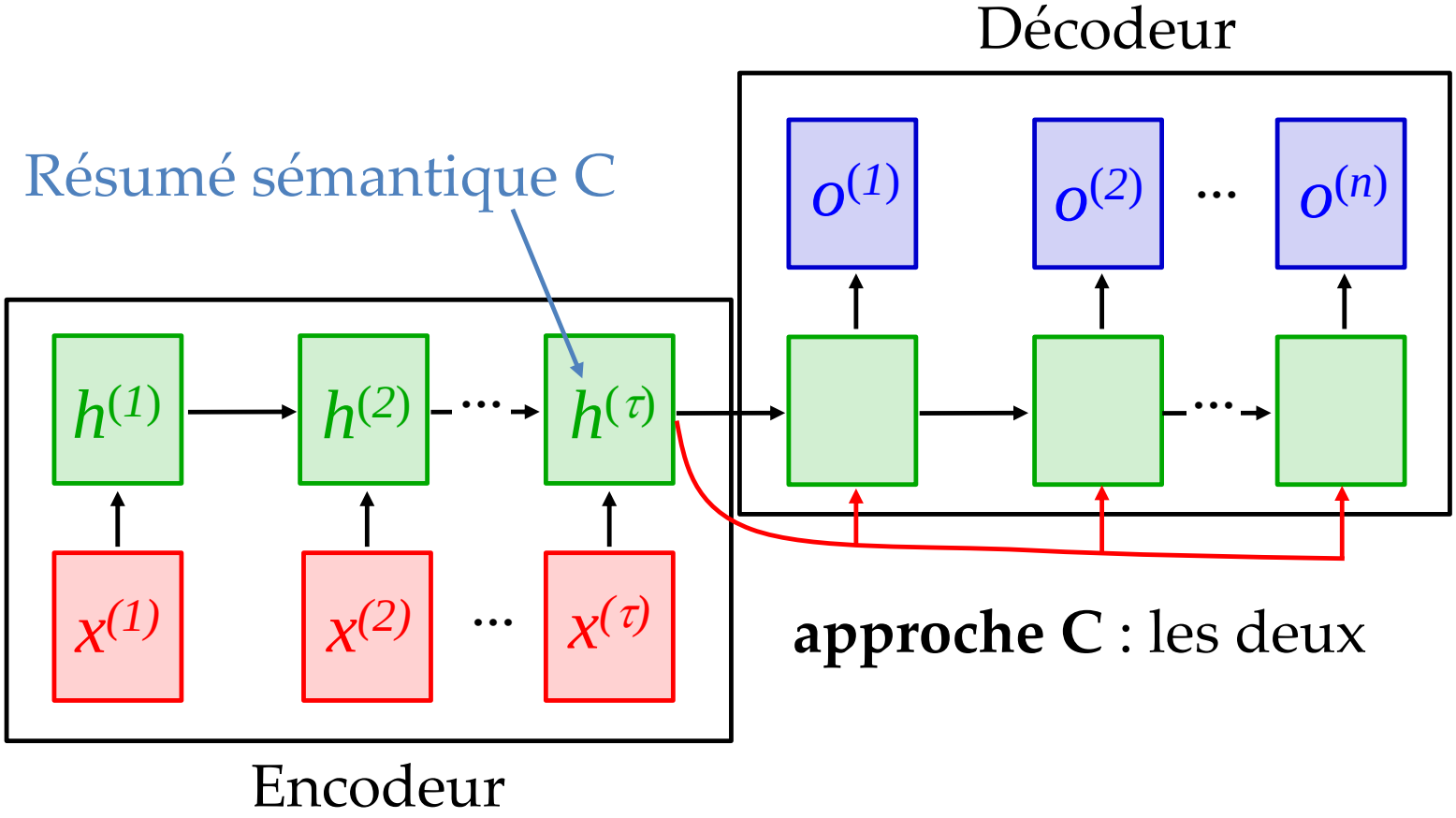
approche A : état $h^{(0)}$ du décodeur

Sequence-to-sequence



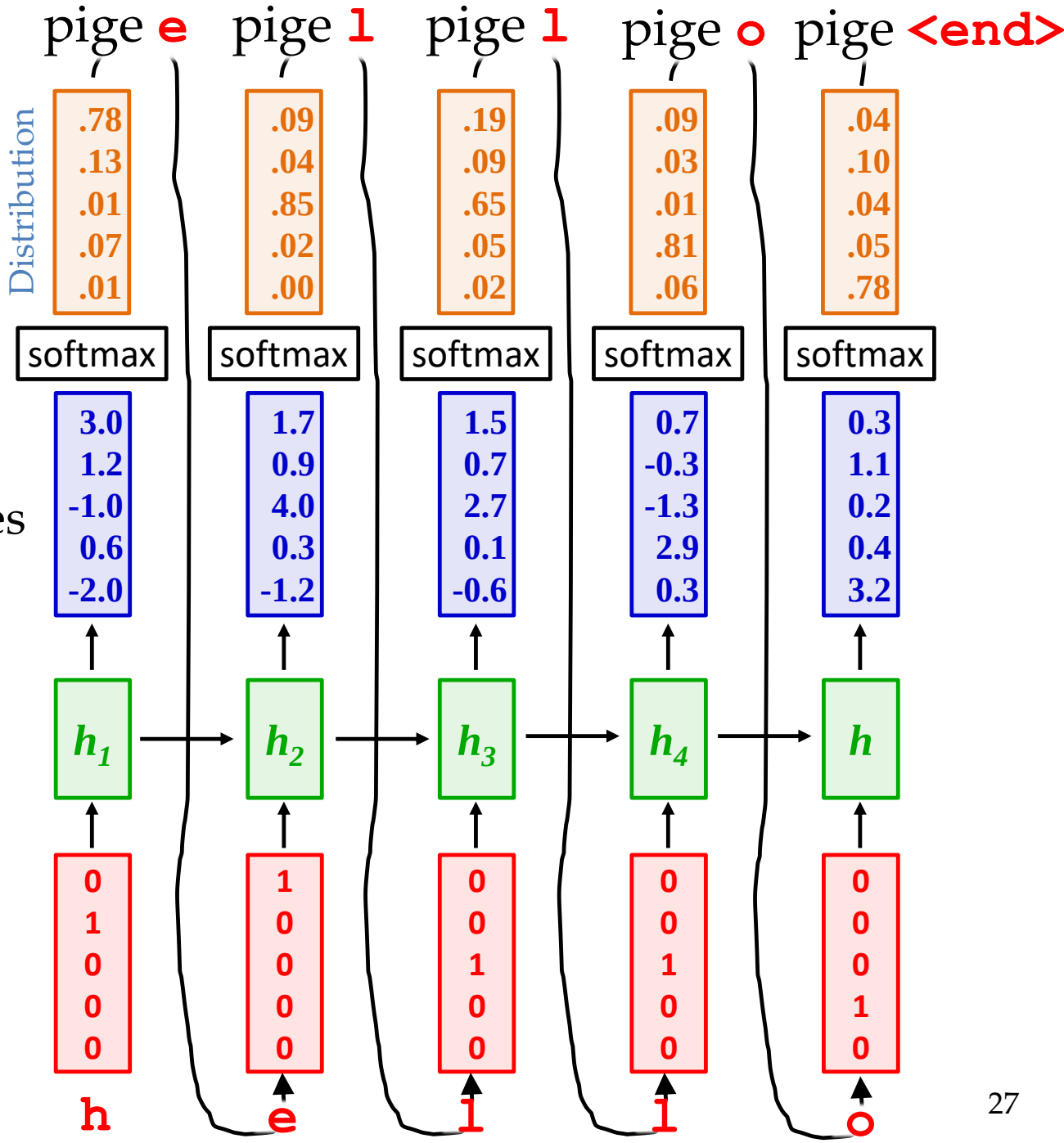
approche B : entrée extra à chaque itération

Sequence-to-sequence



Exemple de génération avec RNN

- Réseau entraîné à prédire des caractères `{e,h,l,o,<end>}`
- Entraîné sur `hello`



adapté de cs231n

Exemple : entraîné sur Shakespeare

- Réseau RNN à trois couches
- 512 neurones cachées par couche
- Entraîné sur 4.4 Mo de données texte

Sortie :

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.
```

http://karpathy.github.io/2015/05/21/rnn-effectiveness/

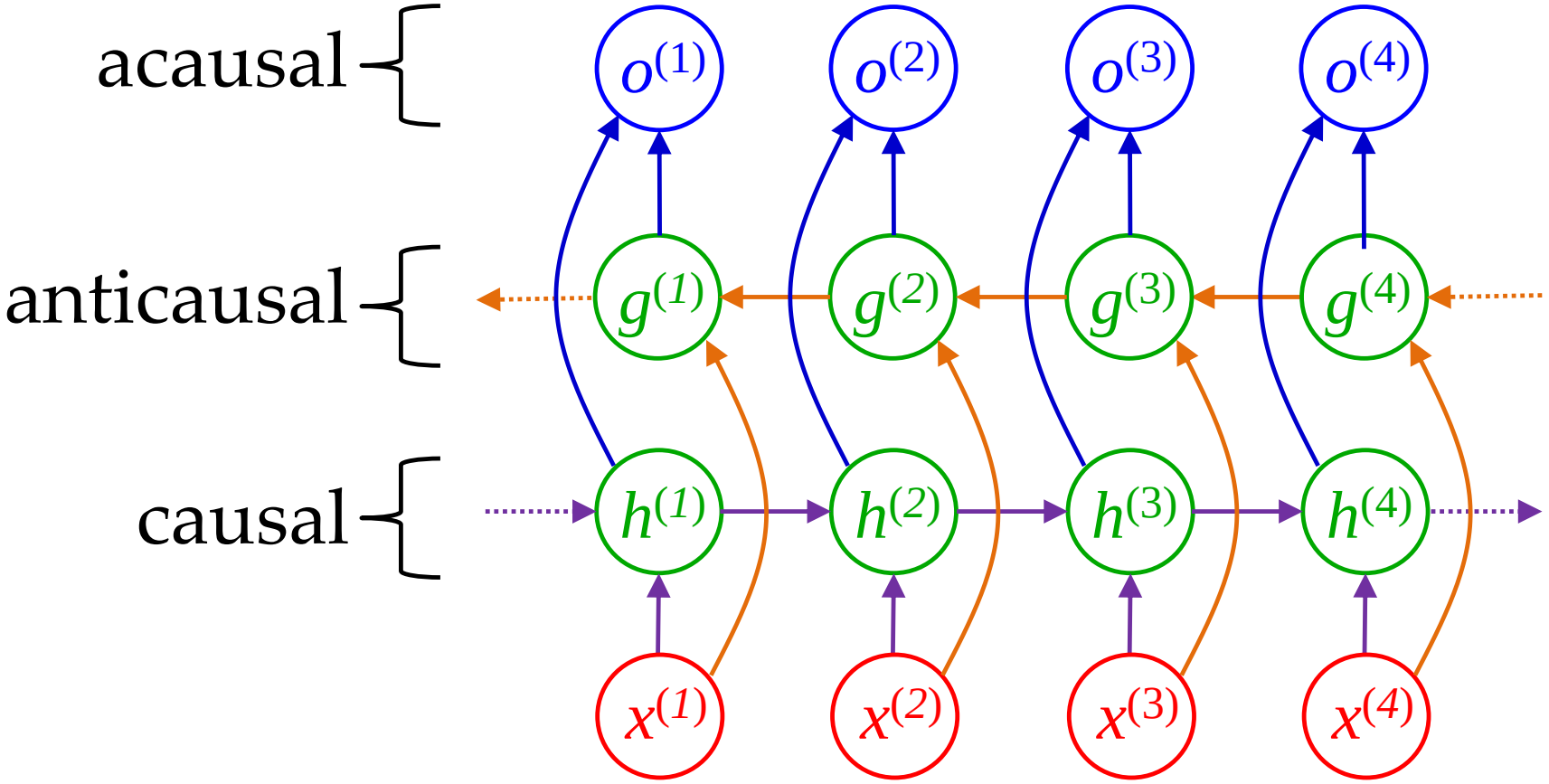
Longueur de sortie $o^{(\tau)}$

- Lors de la génération, on doit s'avoir quand arrêter d'échantillonner le RNN
- **3 stratégies** :
  1. Symbole spécial (**<END>**)
  2. Sortie supplémentaire 0-1 (via sigmoïde), qui prédit la fin
  3. Sortie qui prédit τ directement (régression)

RNN bi-directionnel

- Sortie $o^{(t)}$ peut dépendre de toute la séquence (1 à τ)
- Information pertinente parfois après une entrée x
  - ordre des mots dans une langue
    - adjectif *avant* ou *après* un mot
    - langue SVO, SOV, V2, etc…
  - reconnaissance de la voix
    - coarticulation
  - bio-informatique

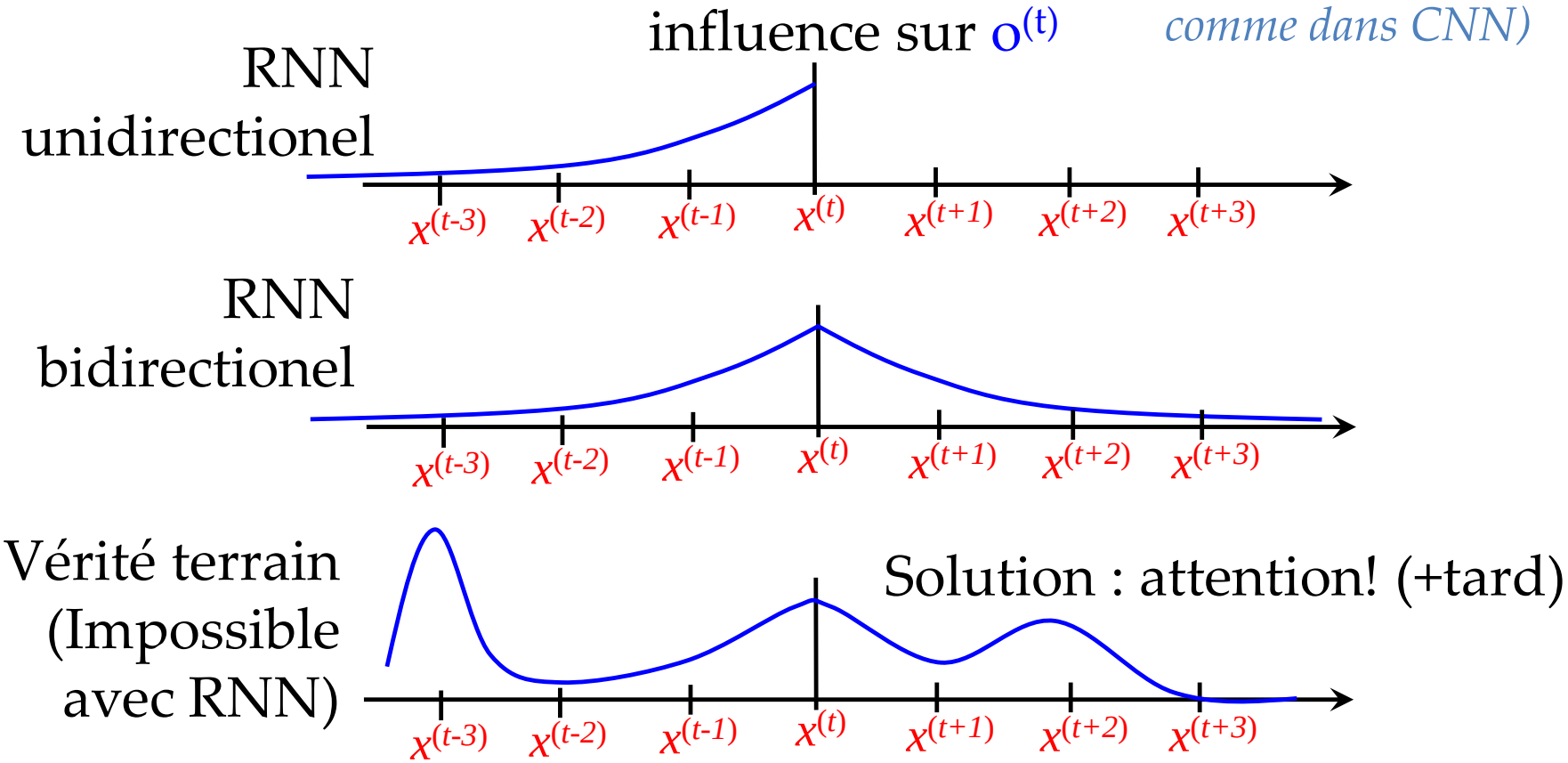
RNN bi-directionnel



Longue portée

- Influence à longue portée difficile dans RNN
- CNN : champ récepteur croissant en profondeur
- RNN : décroissance exponentielle de l'influence

(pas une fenêtre précise, comme dans CNN)

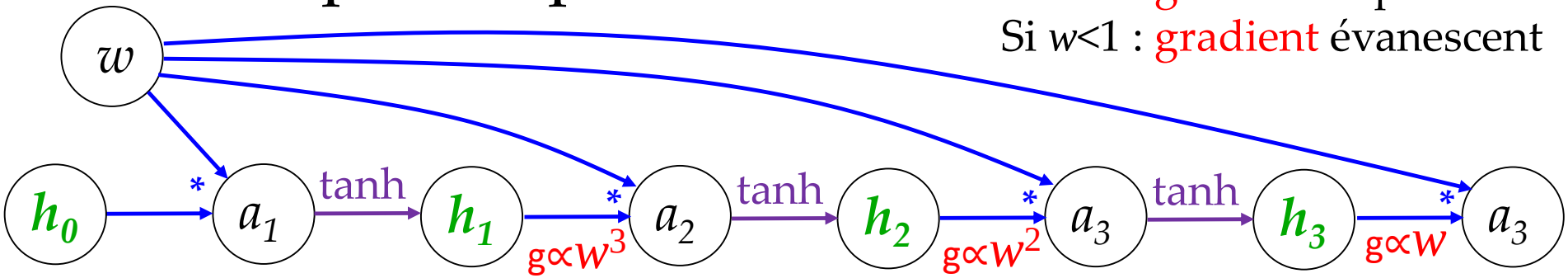


Gradient et entrainement

Exploding/vanishing gradient

- Poids W partagés
- Exemple simplifié :

Si $w>1$: gradient explose
Si $w<1$: gradient évanescent



Pour un réseau récurrent linéaire (simplification) : $h^{(t)} = W^{\mathrm{T}} h^{(t-1)}$

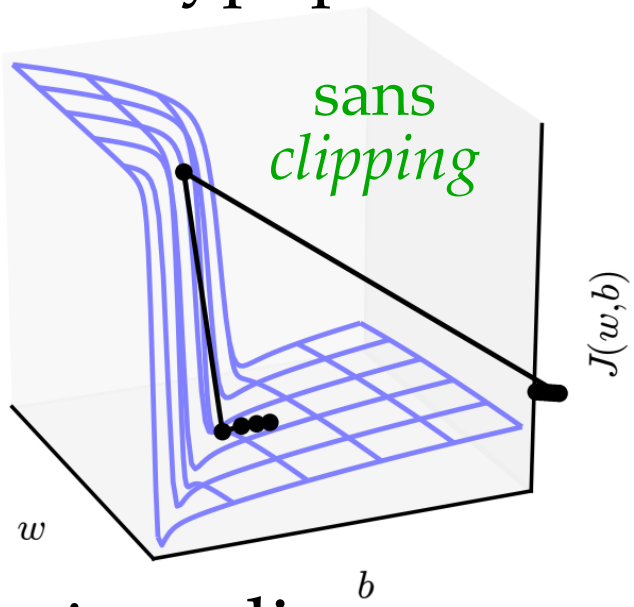
Décomposition en éléments propres $W = Q \Lambda Q^{\mathrm{T}}$

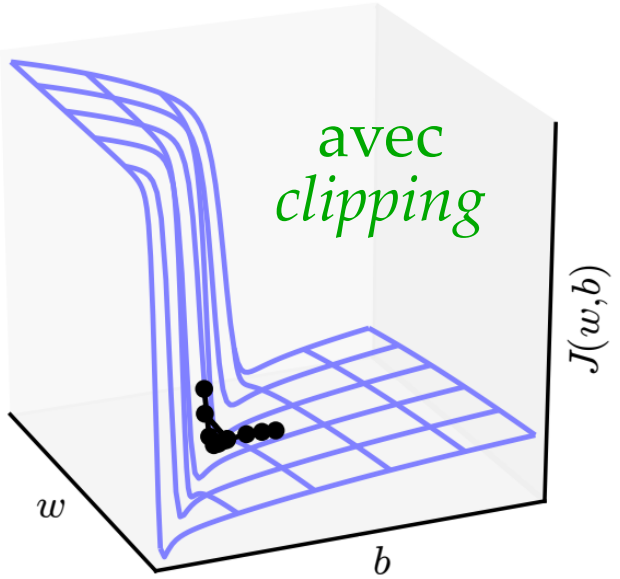
$h^{(t)} = Q^{\mathrm{T}} \Lambda^{t} Q h^{(0)}$ ⇨ si valeur propre $\lambda > 1$: vecteur propre explose

si valeur propre $\lambda < 1$: vecteur propre évanescent

Gradient clipping pour entraînement RNN

- Ravins typique dans les RNN :





- Solution, clipper :

la norme du gradient

$$if \ \|\vec{g}\| > v$$

$$\vec{g} \frac{v}{\|\vec{g}\|}$$

les entrées du gradient, individuellement

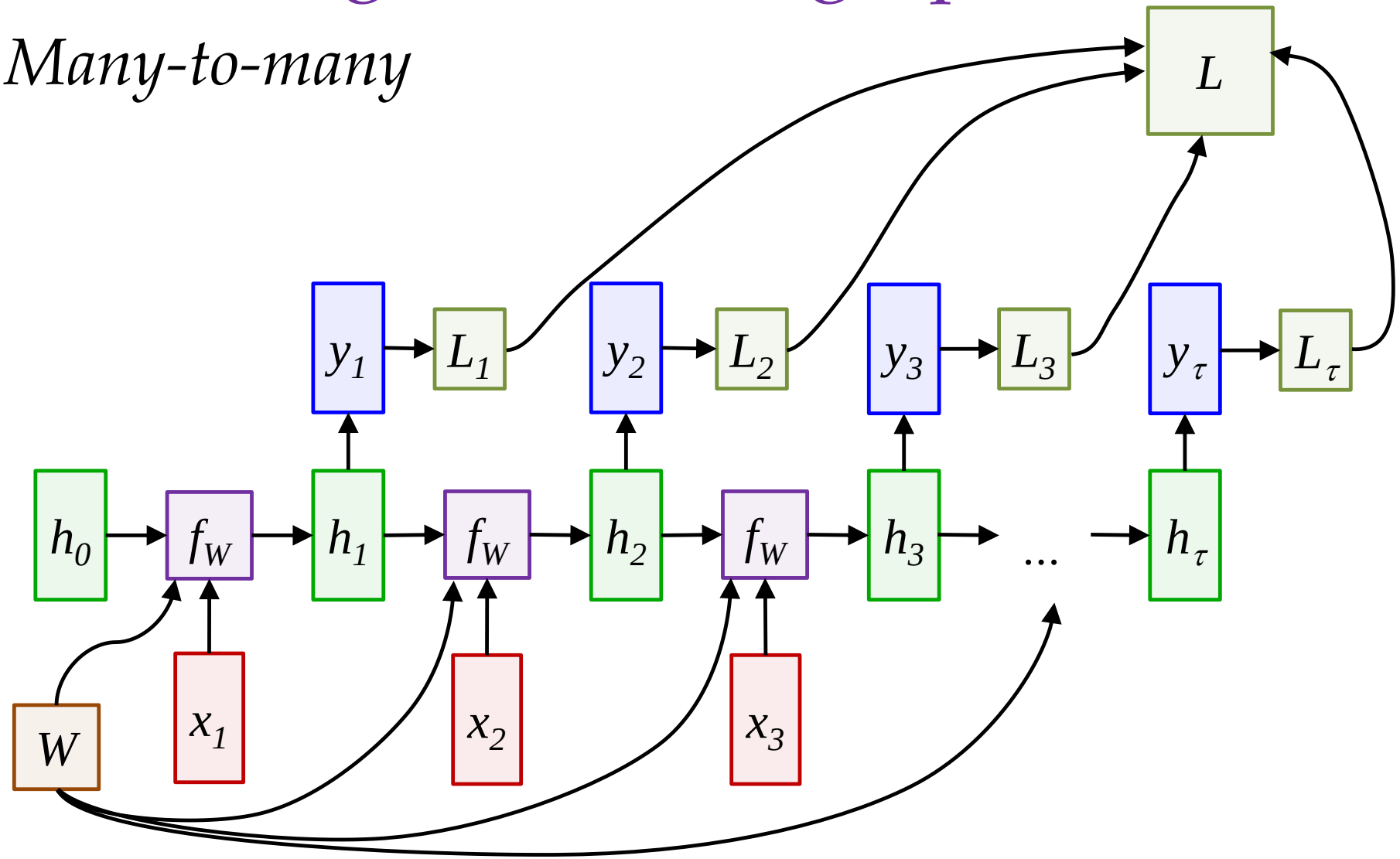
$$\vec{g} = \begin{bmatrix} 88493.4 \\ -0.3 \\ \dots \\ -9948423 \end{bmatrix} = \begin{bmatrix} v \\ -0.3 \\ \dots \\ -v \end{bmatrix}$$

(similaire comme scaling par paramètre, lors de l'optimisation)

- Si NaN, bouger au hasard d'une magnitude v

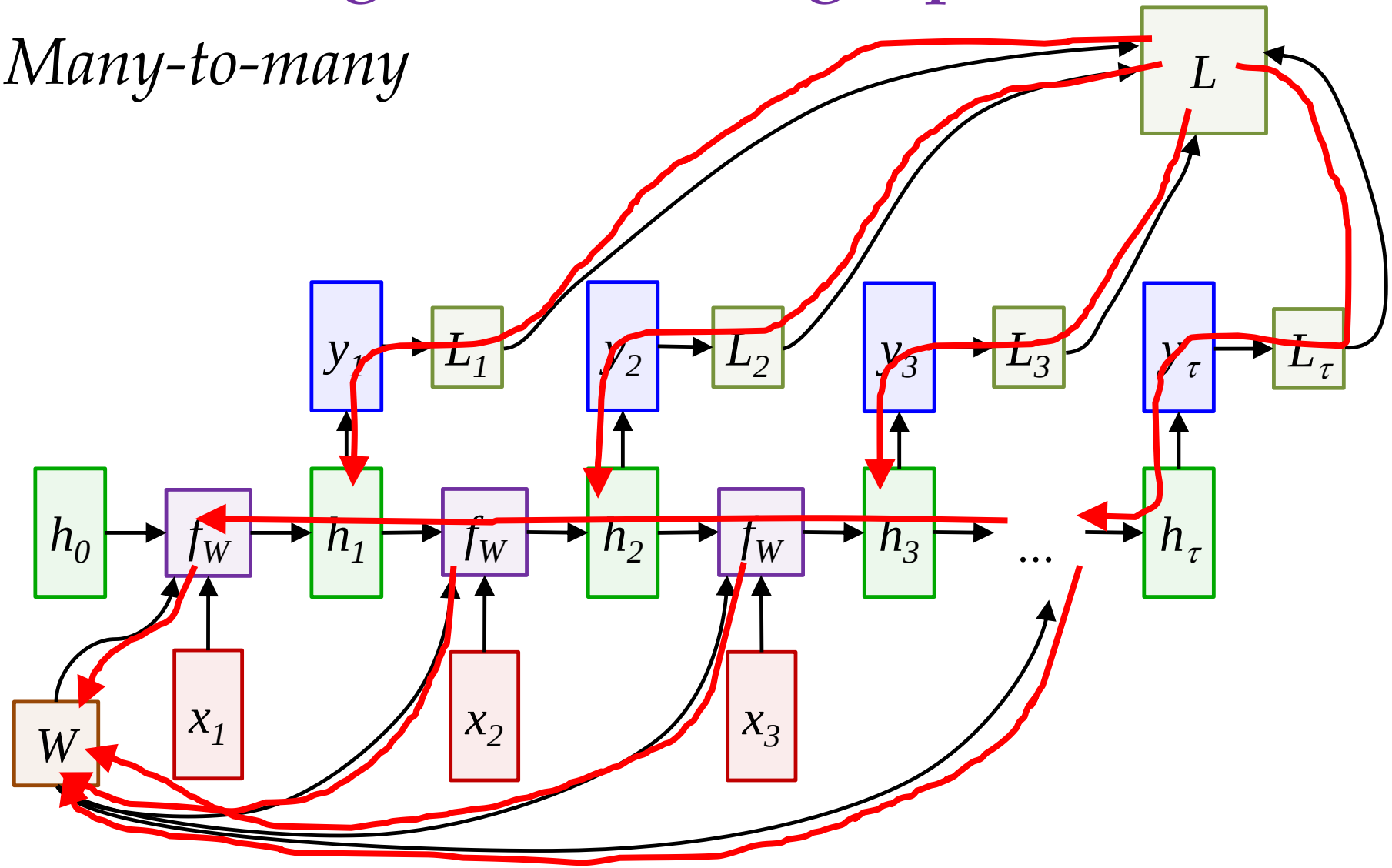
tiré de : I. Goodfellow et al. Deep Learning

Calcul du gradient sur graphe déroulé



adapté de cs231n

Calcul du gradient sur graphe déroulé

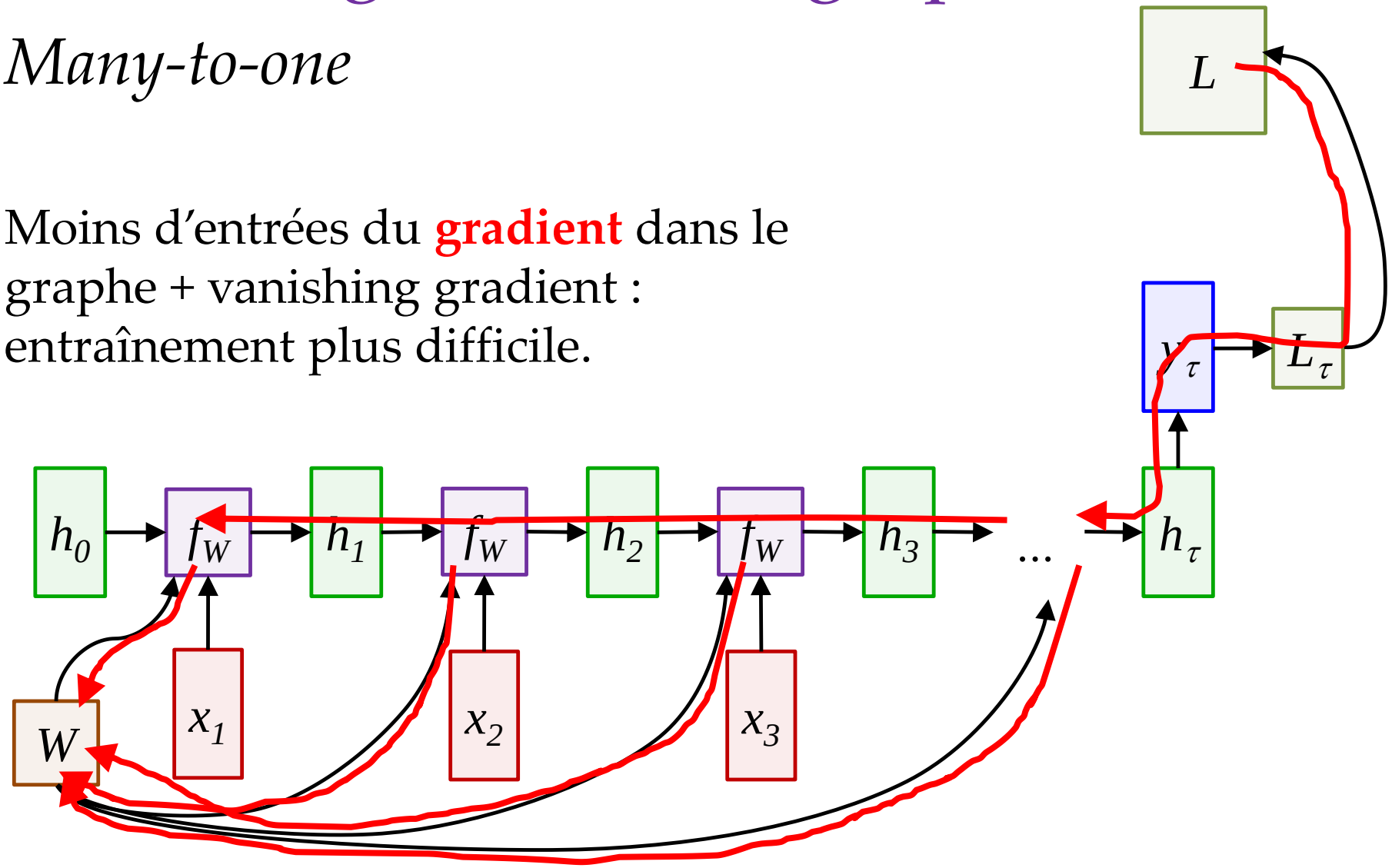


adapté de cs231n

Calcul du gradient sur graphe déroulé

Many-to-one

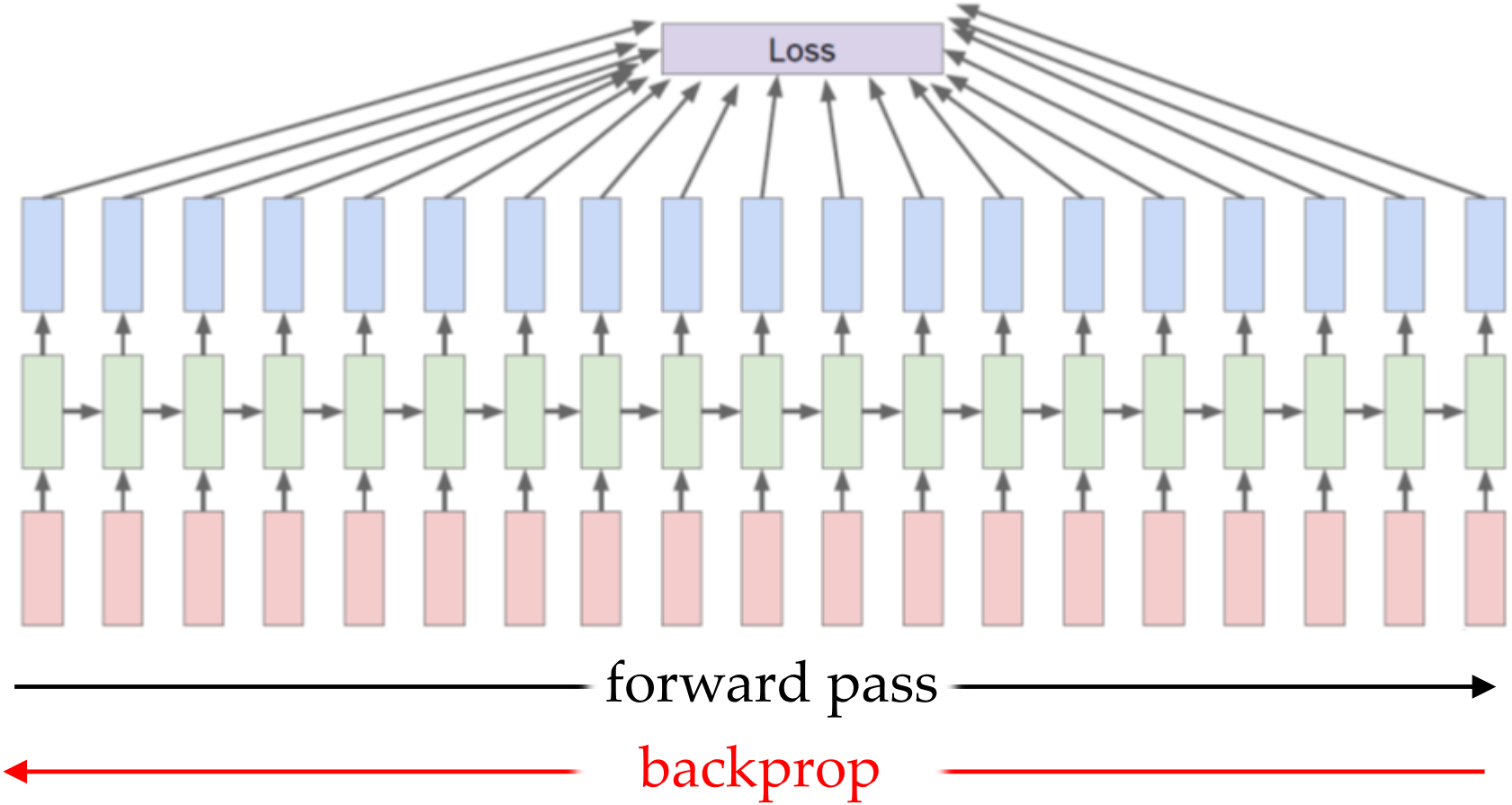
Moins d'entrées du **gradient** dans le graphe + vanishing gradient : entraînement plus difficile.



adapté de cs231n

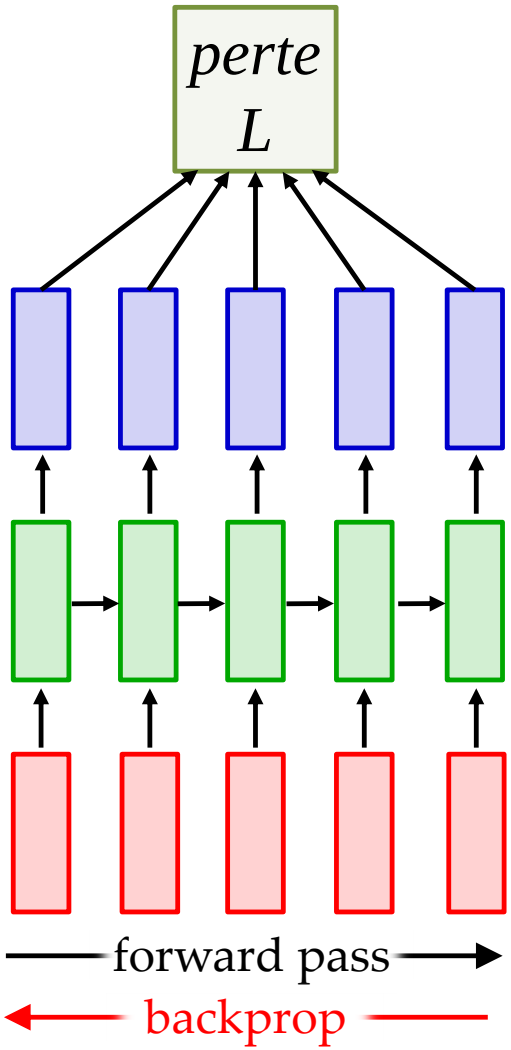
Backprop through time (BPTT)

- Calcule la séquence au complet



adapté de cs231n

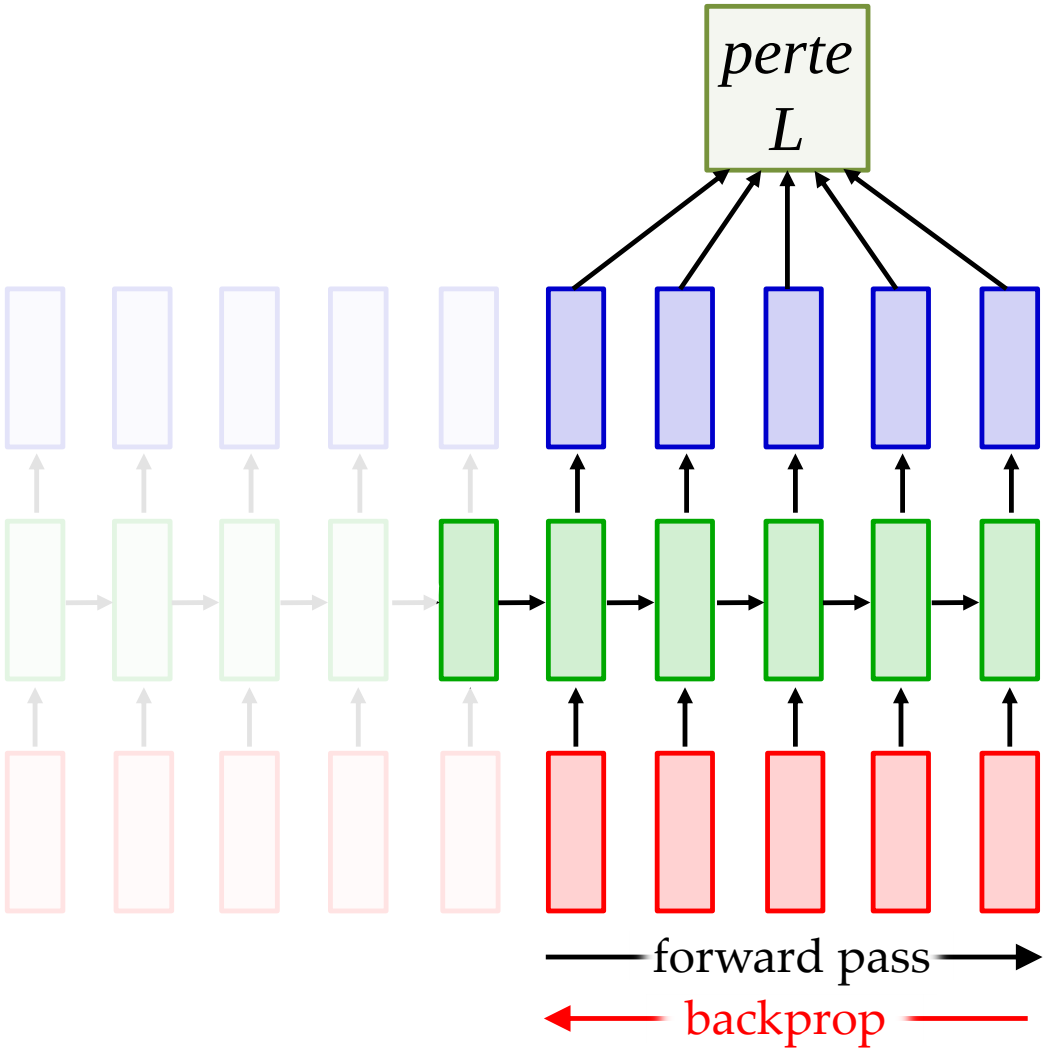
Truncated BPTT



Effectue BPTT sur des segments de la séquence

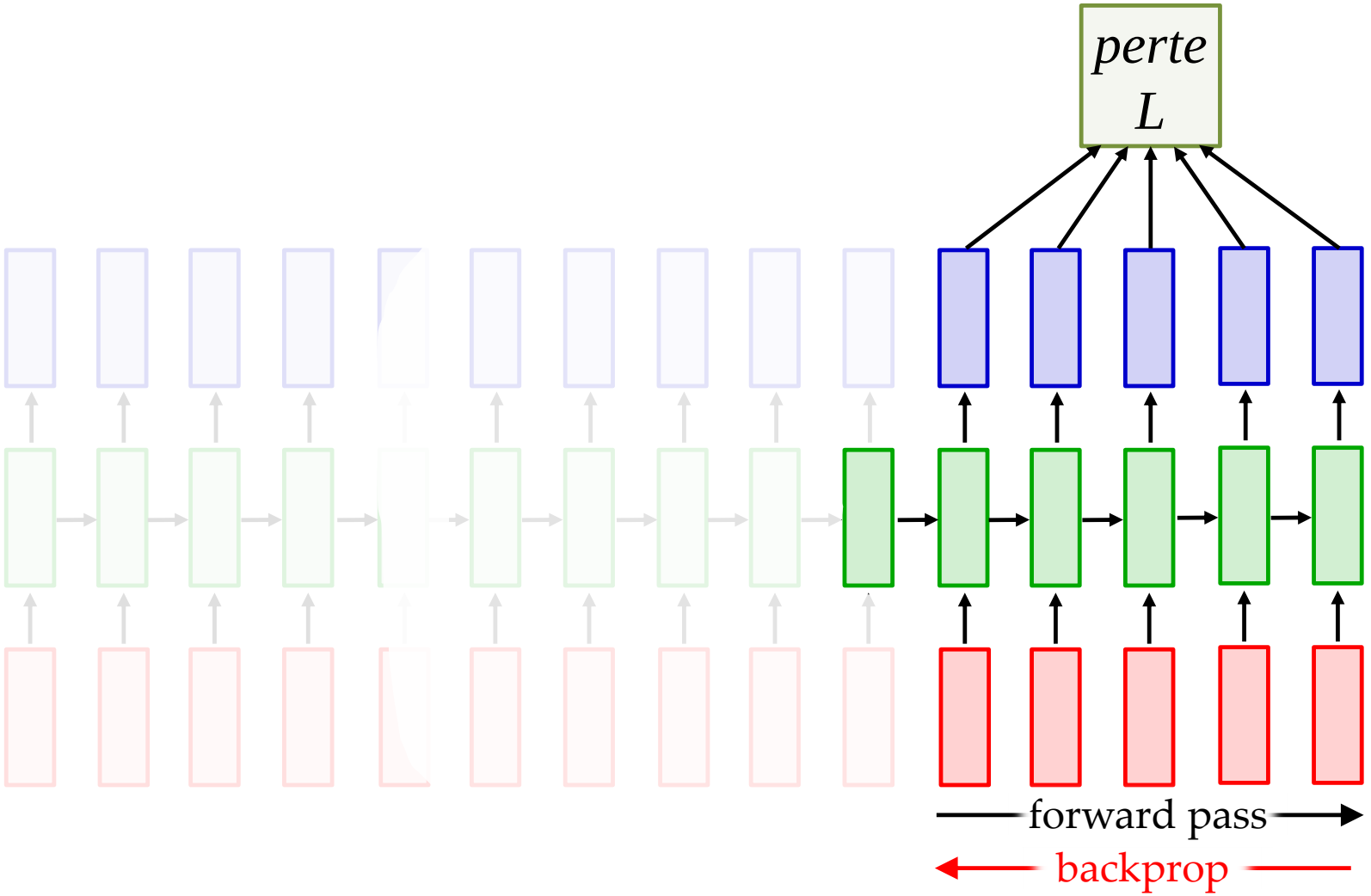
adapté de cs231n

Truncated BPTT



adapté de cs231n

Truncated BPTT



adapté de cs231n