



Bienvenue!

**ÉCOLE D'HIVER FRANCOPHONE
EN APPRENTISSAGE PROFOND**

5 - 9 mars 2018



IVADO

HEC Montréal
Polytechnique Montréal
Université de Montréal



Institut
des algorithmes
d'apprentissage
de Montréal

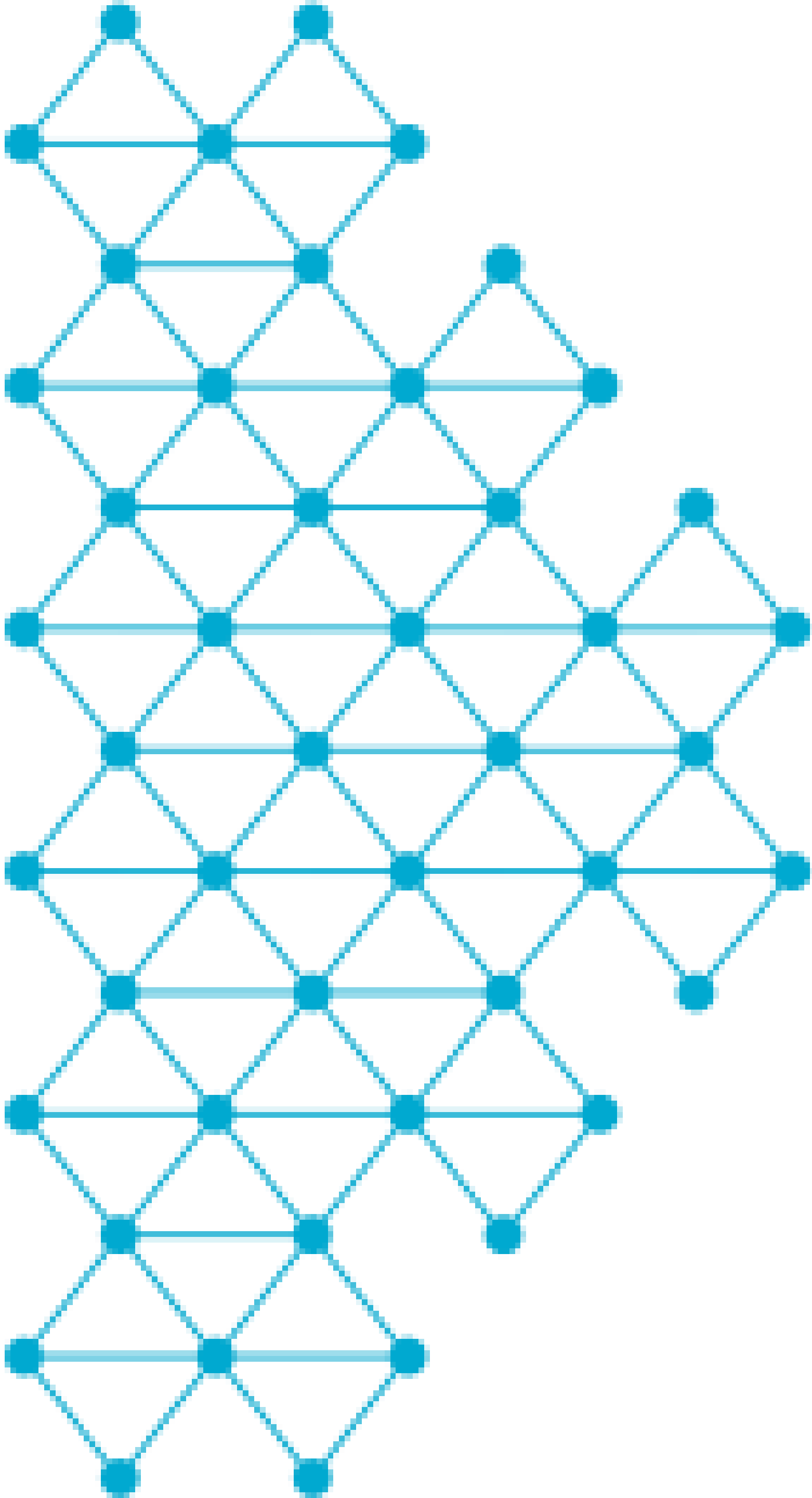


Réseaux Récurrents II

Ecole d'hiver IVADO

César Laurent

- 1. Retour sur la première partie**
- 2. RNNs génératifs**
- 3. Mécanismes d'attention**
- 4. Librairies et références**



1. Retour sur la première partie

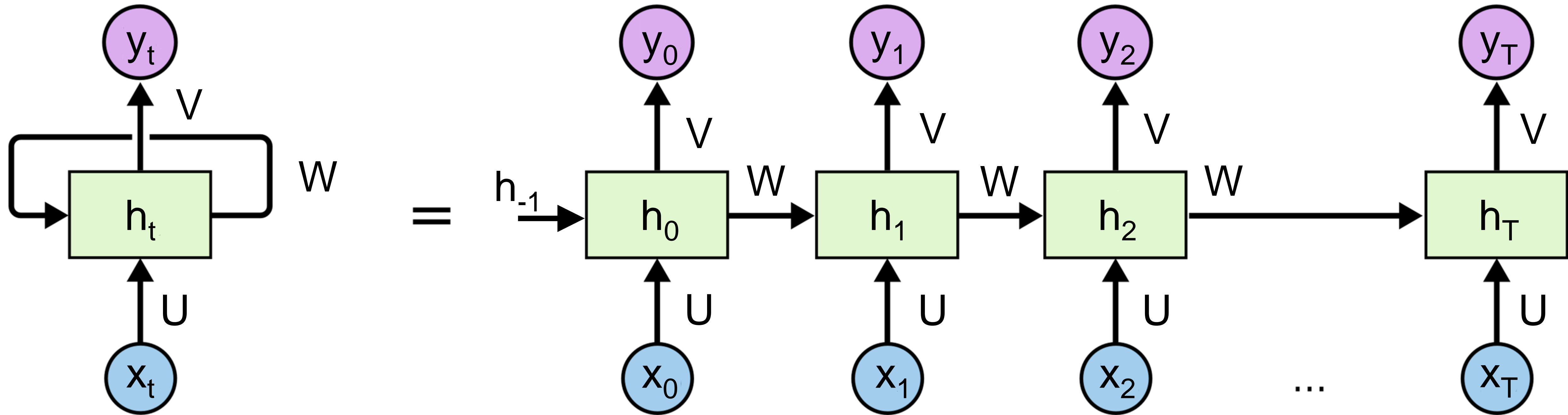
2. RNNs génératifs

3. Mécanismes d'attention

4. Bibliothèques et références

Réseaux Récurrents

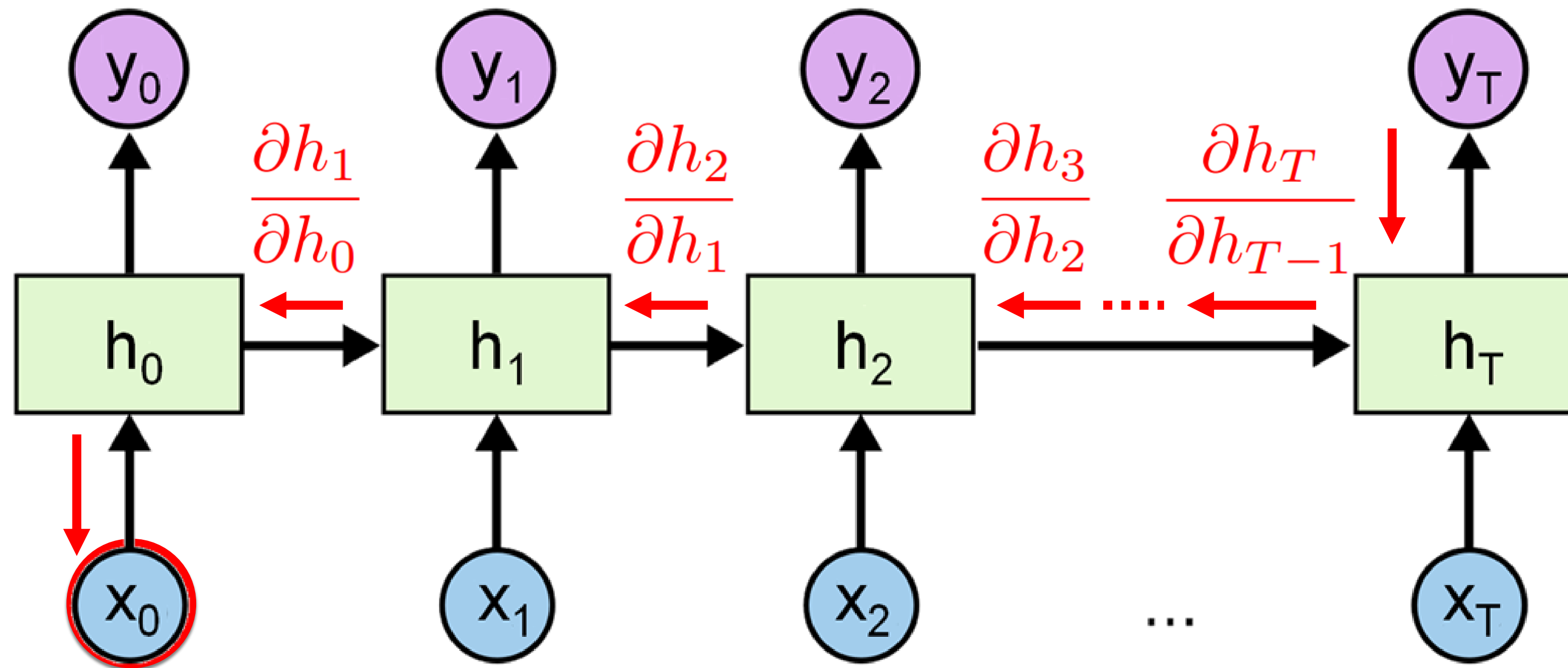
Exemple déroulé dans le temps



Les paramètres sont **partagés** à travers le temps!

Dépendances à long terme

Propagation du gradient



Apprendre des dépendances à long terme

= Propager le gradient loin dans le temps

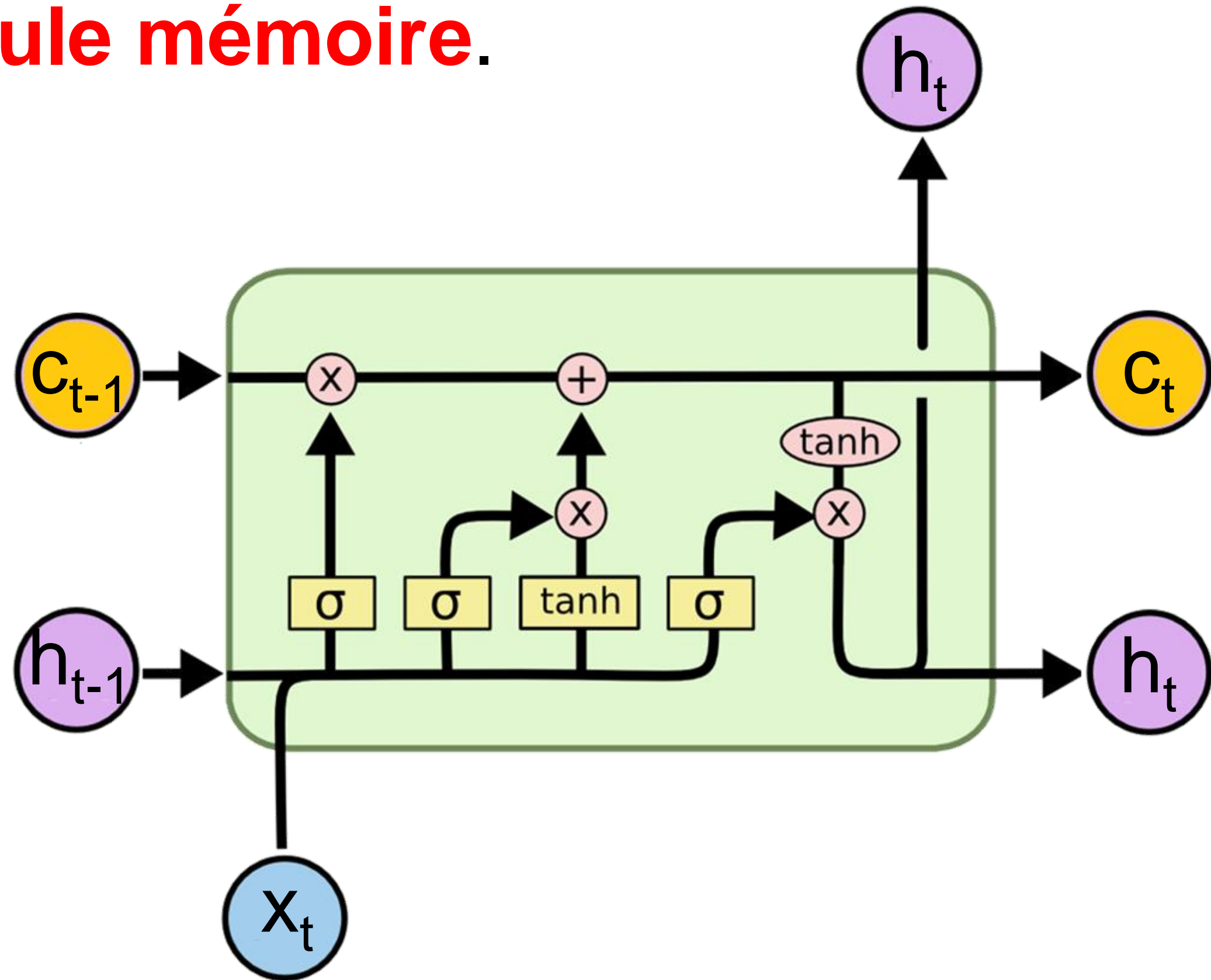
Long Short-Term Memory (LSTM)

En entier

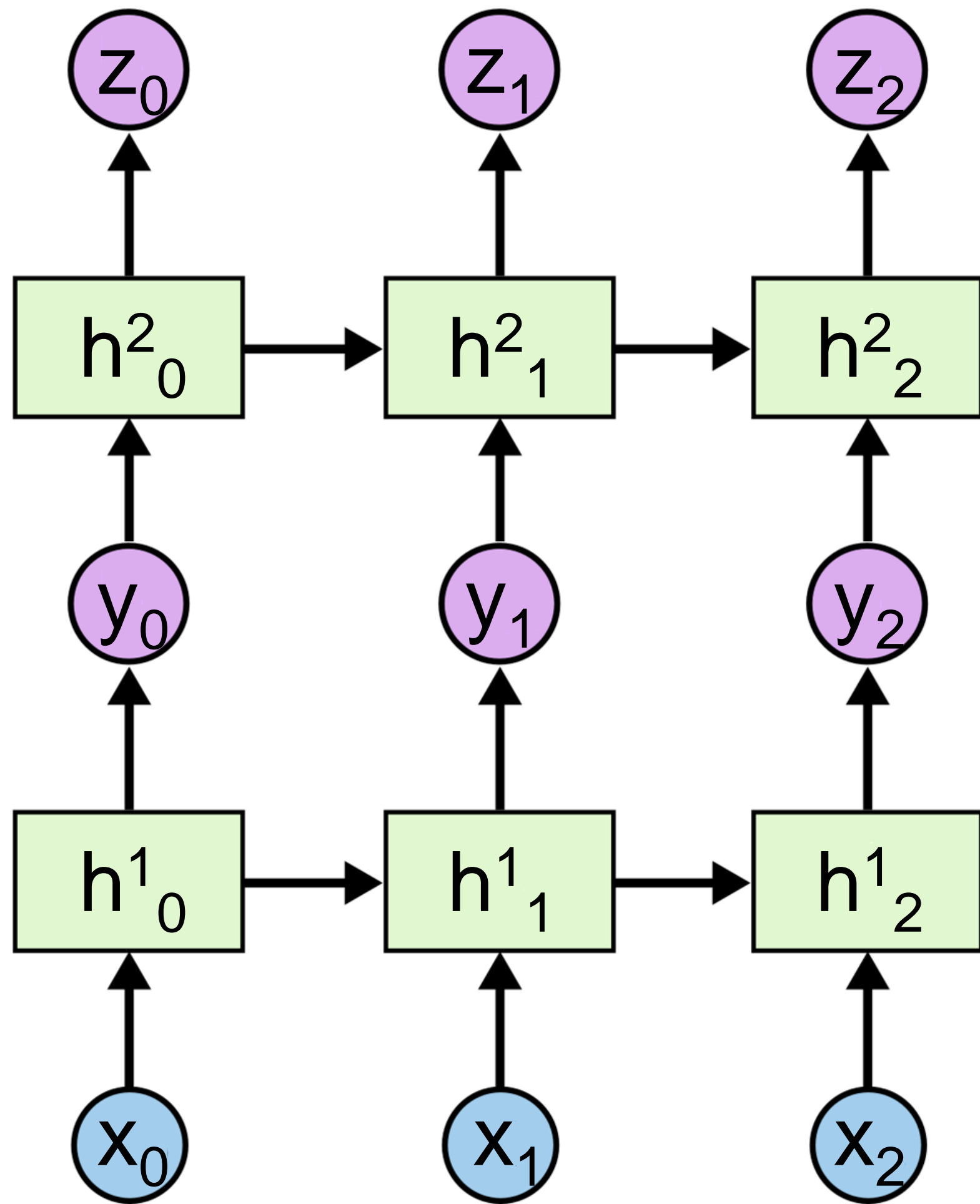


Réduction du problème de dissipation avec **un mécanisme de gates** et une **cellule mémoire**.

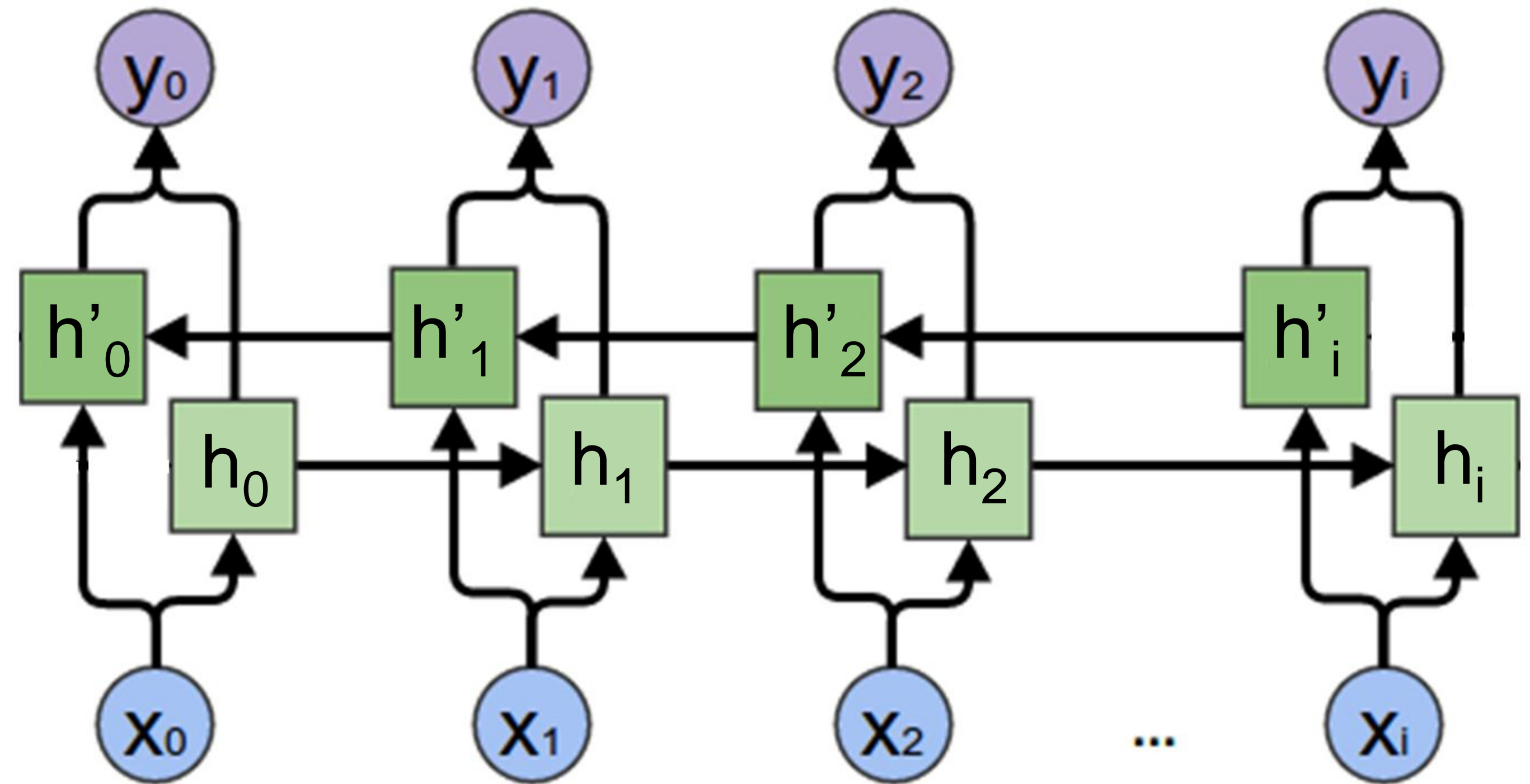
$$\begin{aligned}i_t &= \sigma(U_i x_t + W_i h_{t-1} + b_i) \\f_t &= \sigma(U_f x_t + W_f h_{t-1} + b_f) \\o_t &= \sigma(U_o x_t + W_o h_{t-1} + b_o) \\g_t &= \tanh(U_g x_t + W_g h_{t-1} + b_g) \\c_t &= i_t \odot g_t + f_t \odot c_{t-1} \\h_t &= o_t \odot \tanh(c_t)\end{aligned}$$



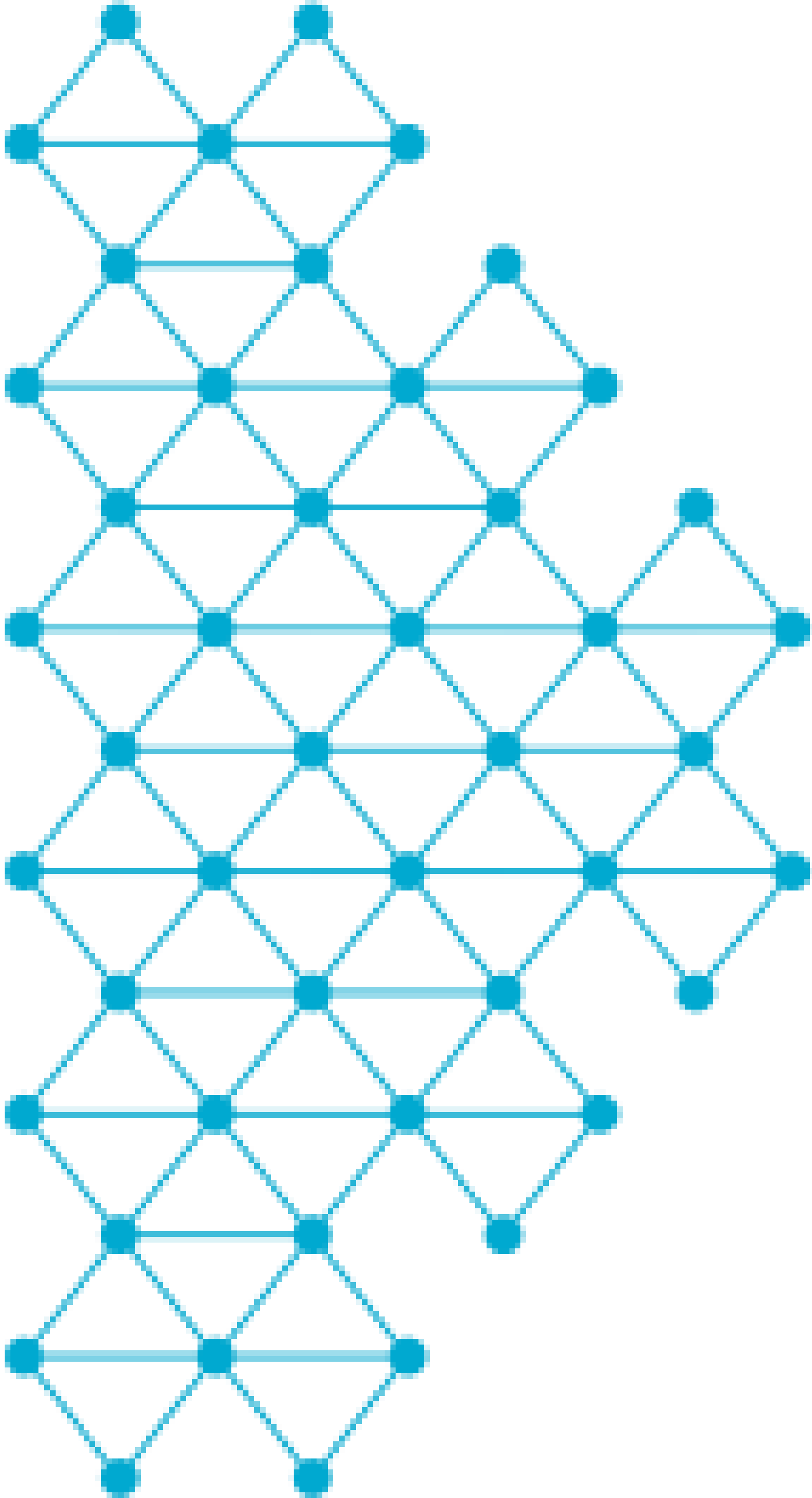
Piles et RNNs Bidirectionnels



Pile de RNNs



RNN Bidirectionnel



1. Retour sur la première partie

2. RNNs génératifs

3. Mécanismes d'attention

4. Bibliothèques et références

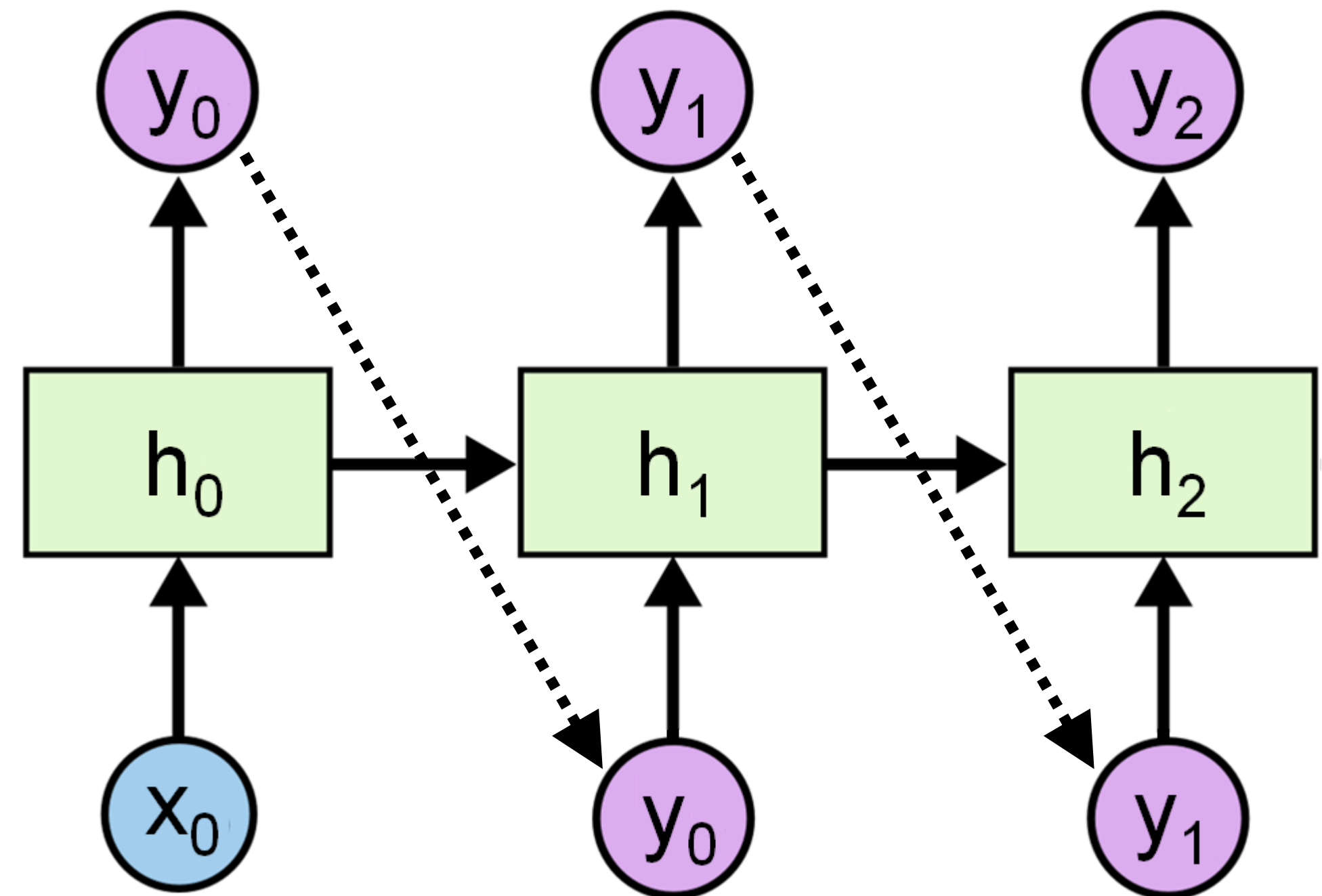
Réseaux Récurrents Génératifs



Introduction

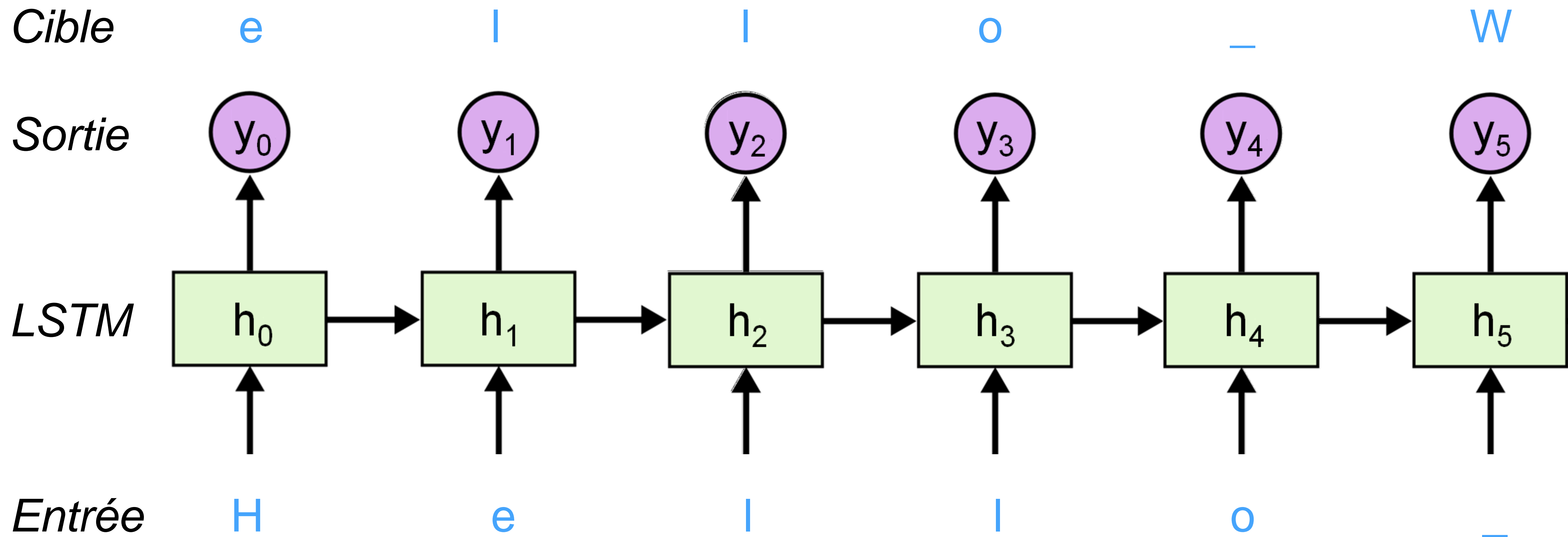
On peut utiliser un RNN pour générer des séquences:

- On donne la sortie au temps t comme entrée au temps $t+1$
- Le modèle génère une séquence lui-même!



Exemple d'utilisation

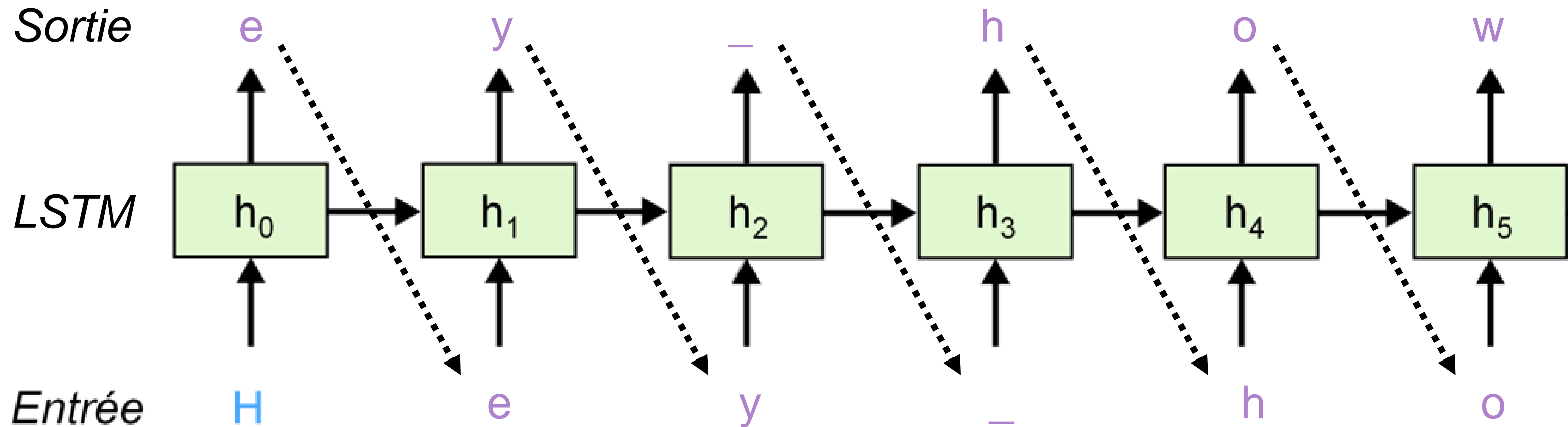
Entraînement d'un modèle de langue (texte)



Entraînement: Prédiction du prochain caractère de la séquence: $c_t = x_{t+1}$

Exemple d'utilisation

Test d'un modèle de langue (texte)



Test: On utilise la sortie au temps t comme entrée au temps $t+1$

Réseaux Récurrents Génératifs



Exemple de génération: Entraîné sur des textes de WSJ

Juste après l'initialisation:

- *« usb9xkrd9ruaias\$dsaj'4lmjwyd61\se.lcn6jey0pbco40ab'65<8um324nqdhm<ufwt#y*/w5bt'nm.zq«2rqm-a2'2mst#u315w&tNwdqNafqh »*

Après la première époque:

- *« to will an apple for a N shares of the practiced to working rudle and a dow listed that scill extressed holding a »*

Après 76 époques:

- *« president economic spokesman executive for securities was support to put used the sharelike the acquired who pla »*

Réseaux Récurrents Génératifs



Exemple de génération: Entraîné sur le code source de Linux

```
/*
 * Increment the size file of the new incorrect UI_FILTER group information
 * of the size generatively.
 */
static int indicate_policy(void)
{
    int error;
    if (fd == MARN_EPT) {
        /*
         * The kernel blank will coeld it to userspace.
         */
        if (ss->segment < mem_total)
            unblock_graph_and_set_blocked();
        else
            ret = 1;
        goto bail;
    }
    segaddr = in_SB(in.addr);
    selector = seg / 16;
    setup_works = true;
    for (i = 0; i < blocks; i++) {
        seq = buf[i++];
        bpf = bd->bd.next + i * search;
        if (fd) {
            current = blocked;
        }
    }
    rw->name = "Getjbbregs";
    bprm_self_clearl(&iv->version);
    regs->new = blocks[(BPF_STATS << info->historidac)] | PFMR_CLOBATHINC_SECONDS << 12;
    return segtable;
}
```

Le RNN a appris la syntaxe du C:

- Commentaires
- Mots-clés
- Ouverture et fermeture () { }
- ...

Mais ne se rappelle pas des variables!

Réseaux Récurrents Génératifs



Exemple de génération: Entraîné sur le code source de Linux

```
/*
 * Copyright (c) 2006-2010, Intel Mobile Communications. All rights reserved.
 *
 * This program is free software; you can redistribute it and/or modify it
 * under the terms of the GNU General Public License version 2 as published by
 * the Free Software Foundation.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
 */

#include <linux/kexec.h>
#include <linux/errno.h>
#include <linux/io.h>
#include <linux/platform_device.h>
#include <linux/multi.h>
#include <linux/ckevent.h>

#include <asm/io.h>
#include <asm/prom.h>
#include <asm/e820.h>
#include <asm/system_info.h>
#include <asm/setew.h>
#include <asm/pgproto.h>

#define REG_PG    vesa_slot_addr_pack
#define PFM_NOCOMP AFSR(0, load)
#define STACK_DDR(type)    (func)
```

De temps en temps le RNN décide qu'il est temps de commencer un nouveau fichier:

- Licence
- Imports
- Macros

Réseaux Récurrents Génératifs

Exemple de génération: Entraîné sur de la musique



- Génération de musique

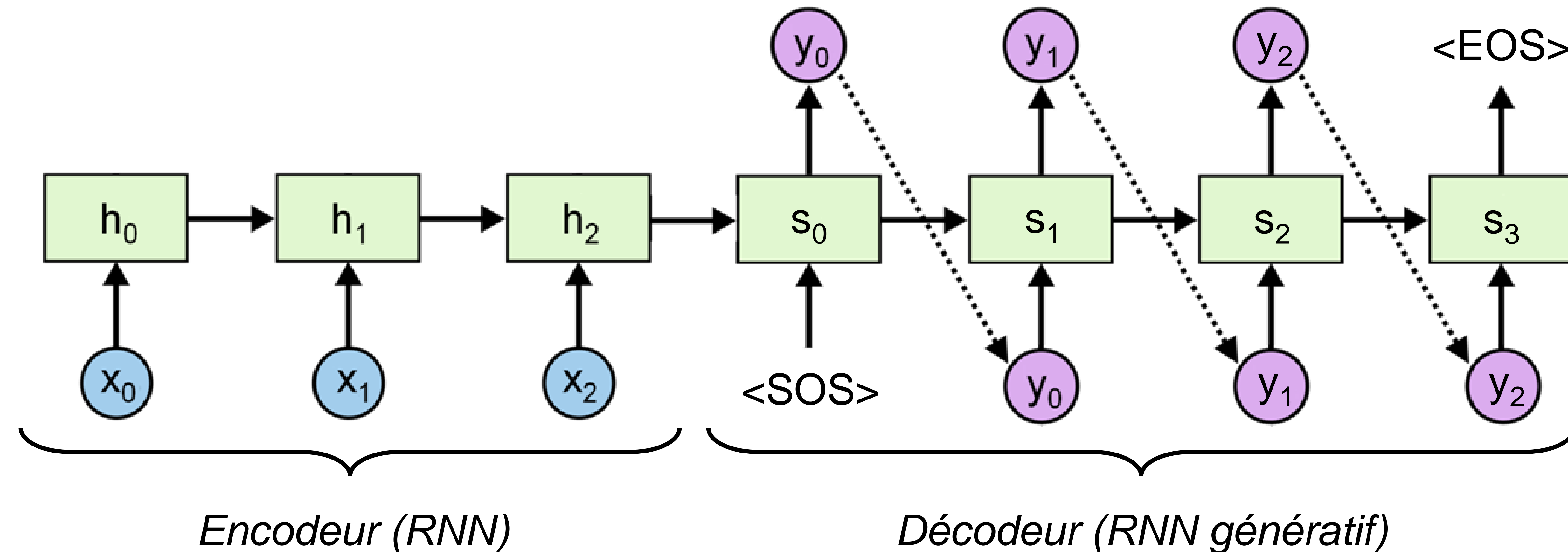
Bon c'est bien joli, mais ça sert à quoi?

Modèle « seq2seq »

Présentation

<SOS> Start of sequence

<EOS> End of sequence



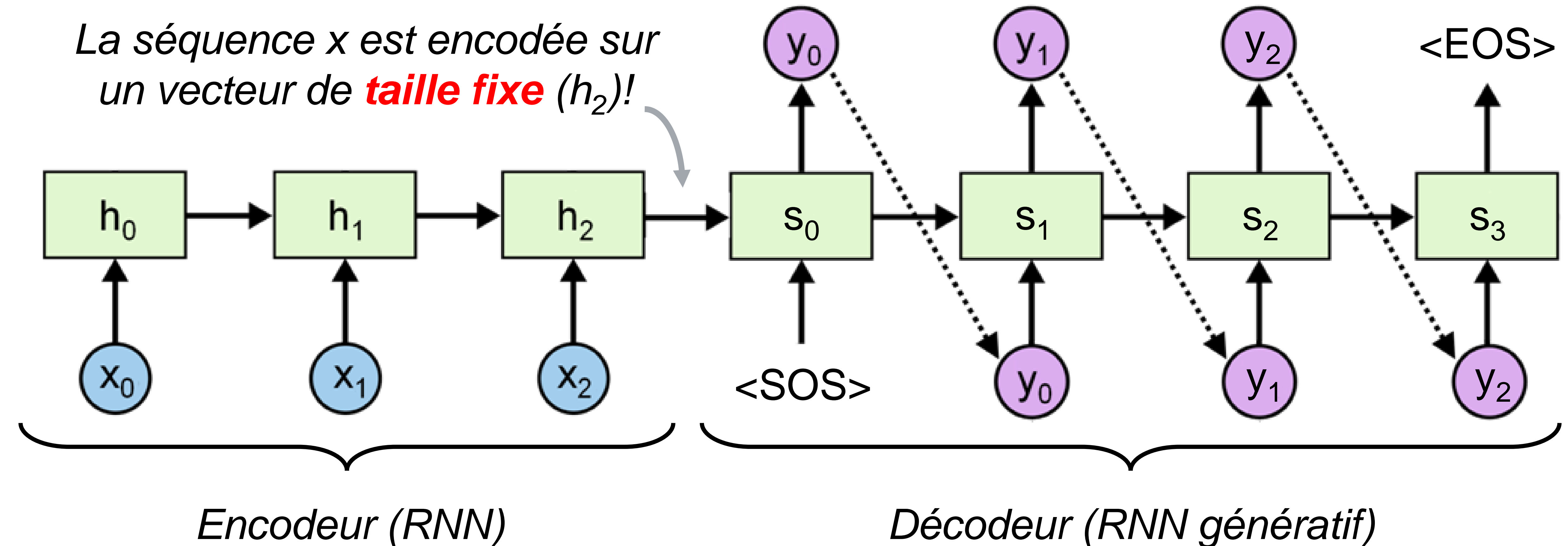
Modèle « seq2seq »

Taille fixe!

<SOS> Start of sequence

<EOS> End of sequence

La séquence x est encodée sur un vecteur de **taille fixe** (h_2)!



Modèle « seq2seq »



Exemple: traduction français-anglais; entraînement

<SOS> Start of sequence

<EOS> End of sequence

Cible: /

am

sick

<EOS>

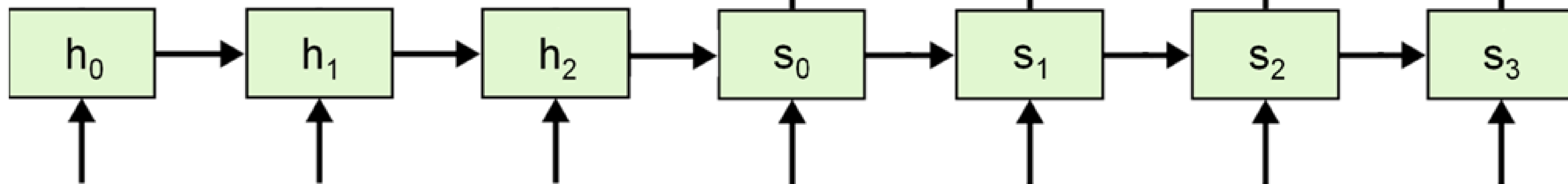
Prédiction y:

y_0

y_1

y_2

y_3



Je

suis

malade

<SOS>

/

am

sick

Encodeur (RNN)

Décodeur (RNN génératif)

Modèle « seq2seq »

Exemple: traduction français-anglais; prédiction

<SOS> Start of sequence

<EOS> End of sequence

Cible: /

am

sick

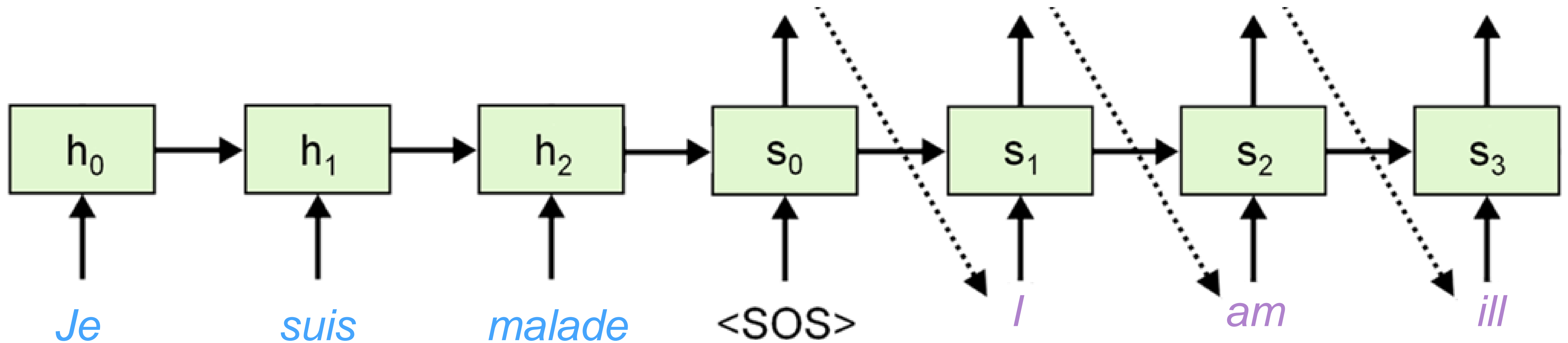
<EOS>

Prédiction y: /

am

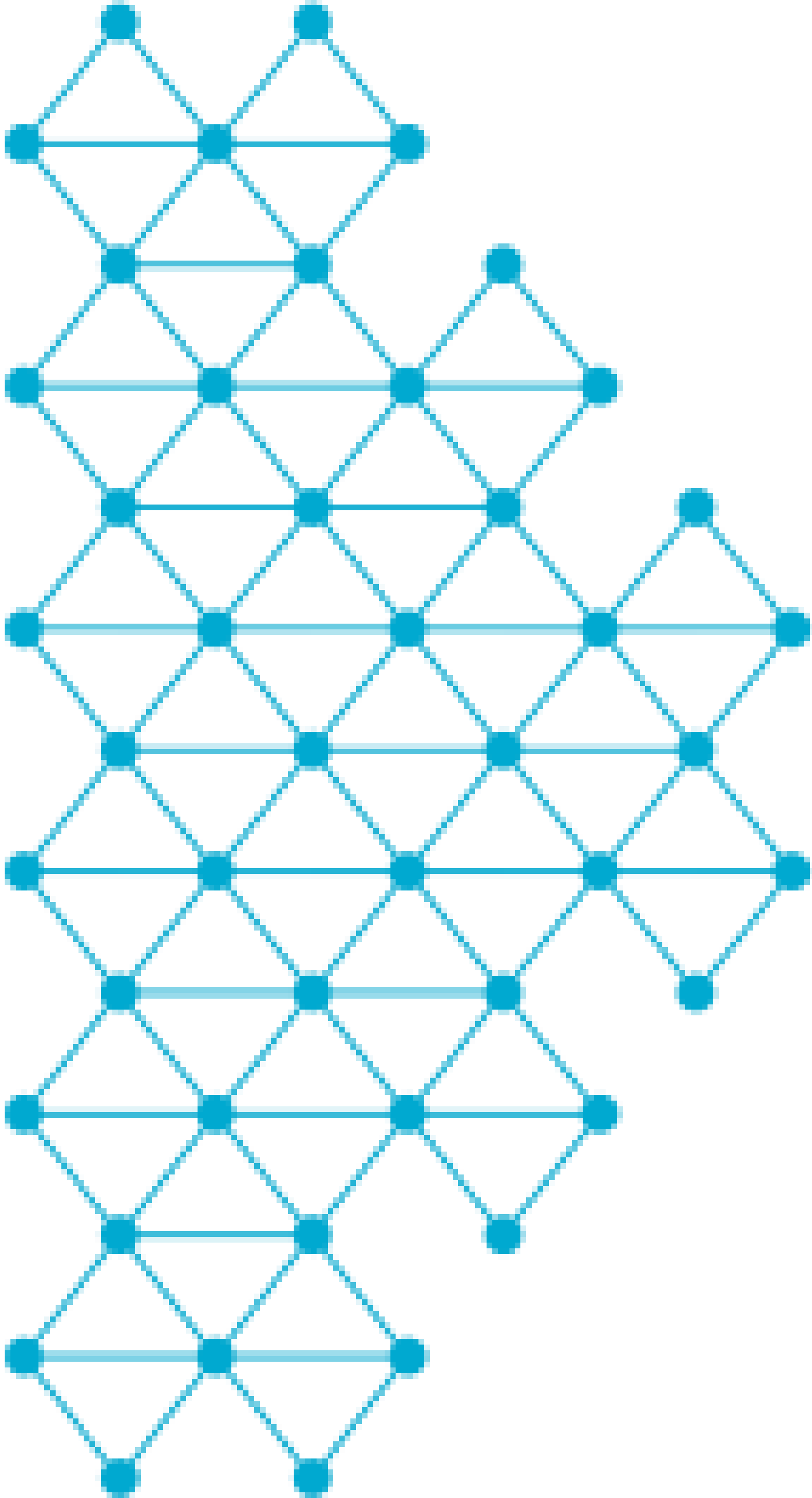
ill

<EOS>



Encodeur (RNN)

Décodeur (RNN génératif)



1. Retour sur la première partie

2. RNNs génératifs

3. Mécanismes d'attention

4. Bibliothèques et références

Mécanisme d'attention

Introduction



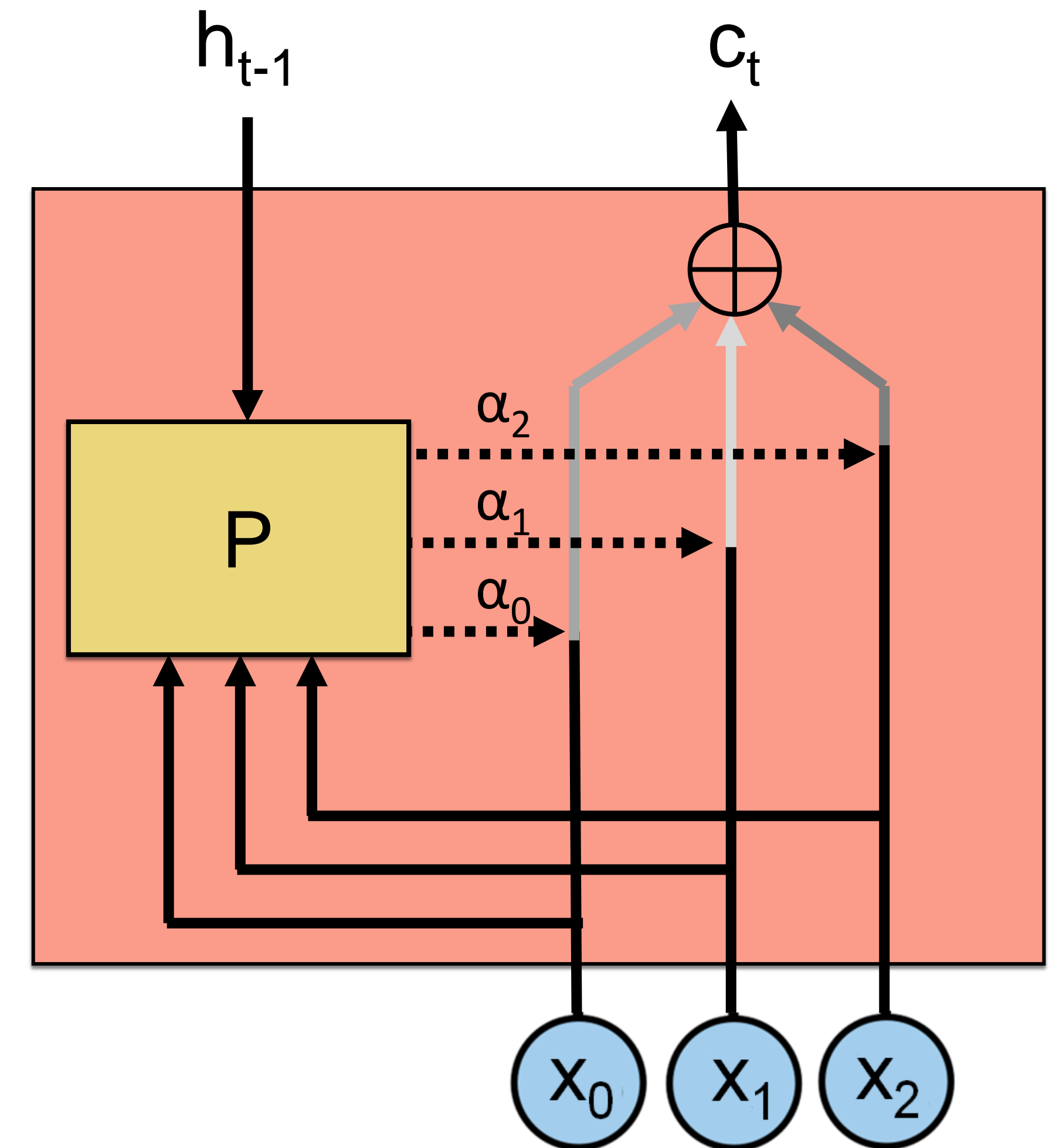
Mécanismes d'attention

Architecture



Un mécanisme d'attention permet à un RNN de porter son attention sur différentes parties de son entrée $[x_0, \dots, x_T]$.

1. Chaque élément de l'entrée est comparé à l'état courant h_{t-1} , pour générer des coefficients de pondération α_i .
2. La sortie c_t (contexte) est une somme pondérée de tous les éléments de l'entrée (« Soft Attention »).

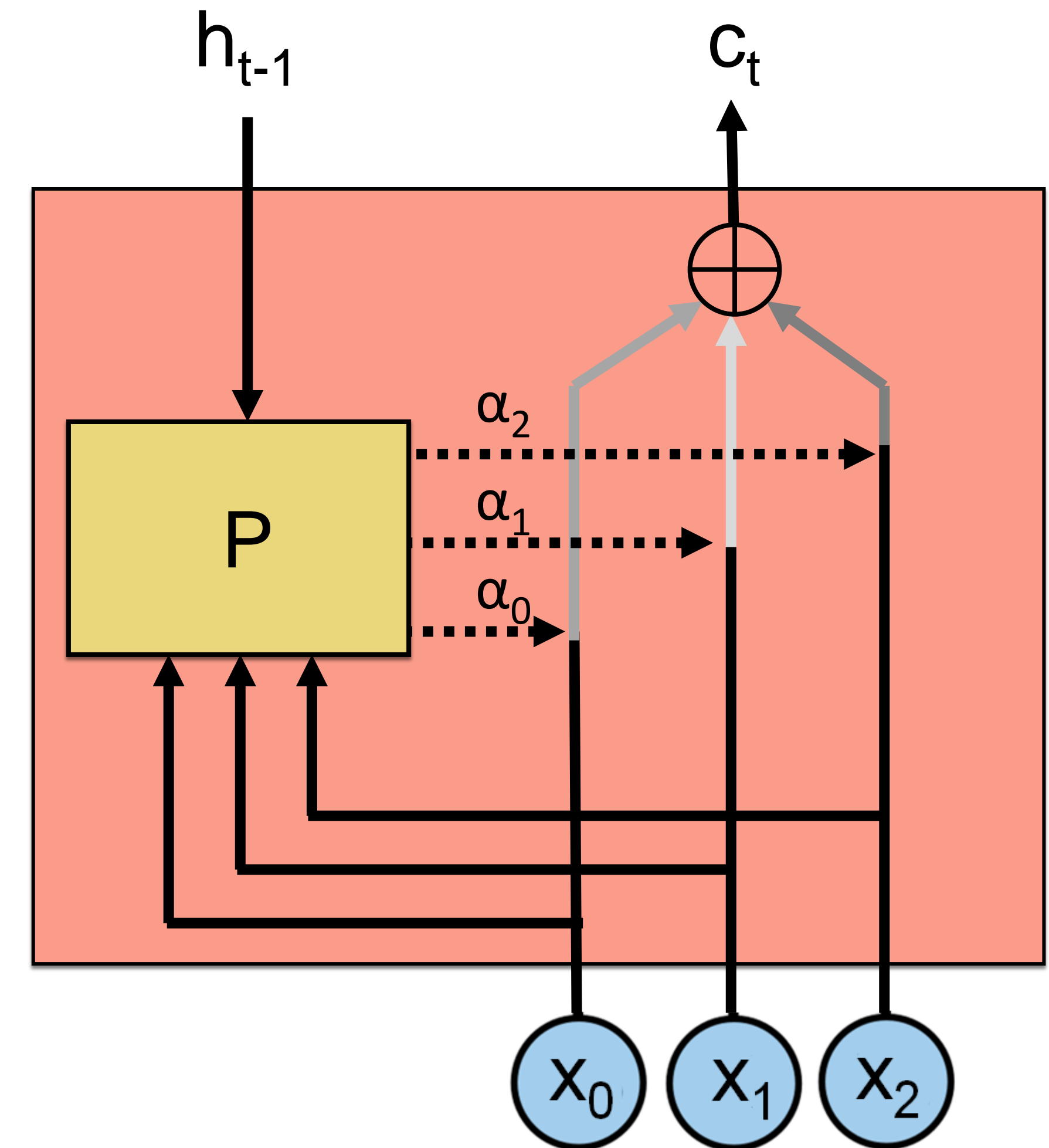


Mécanismes d'attention

Description de P

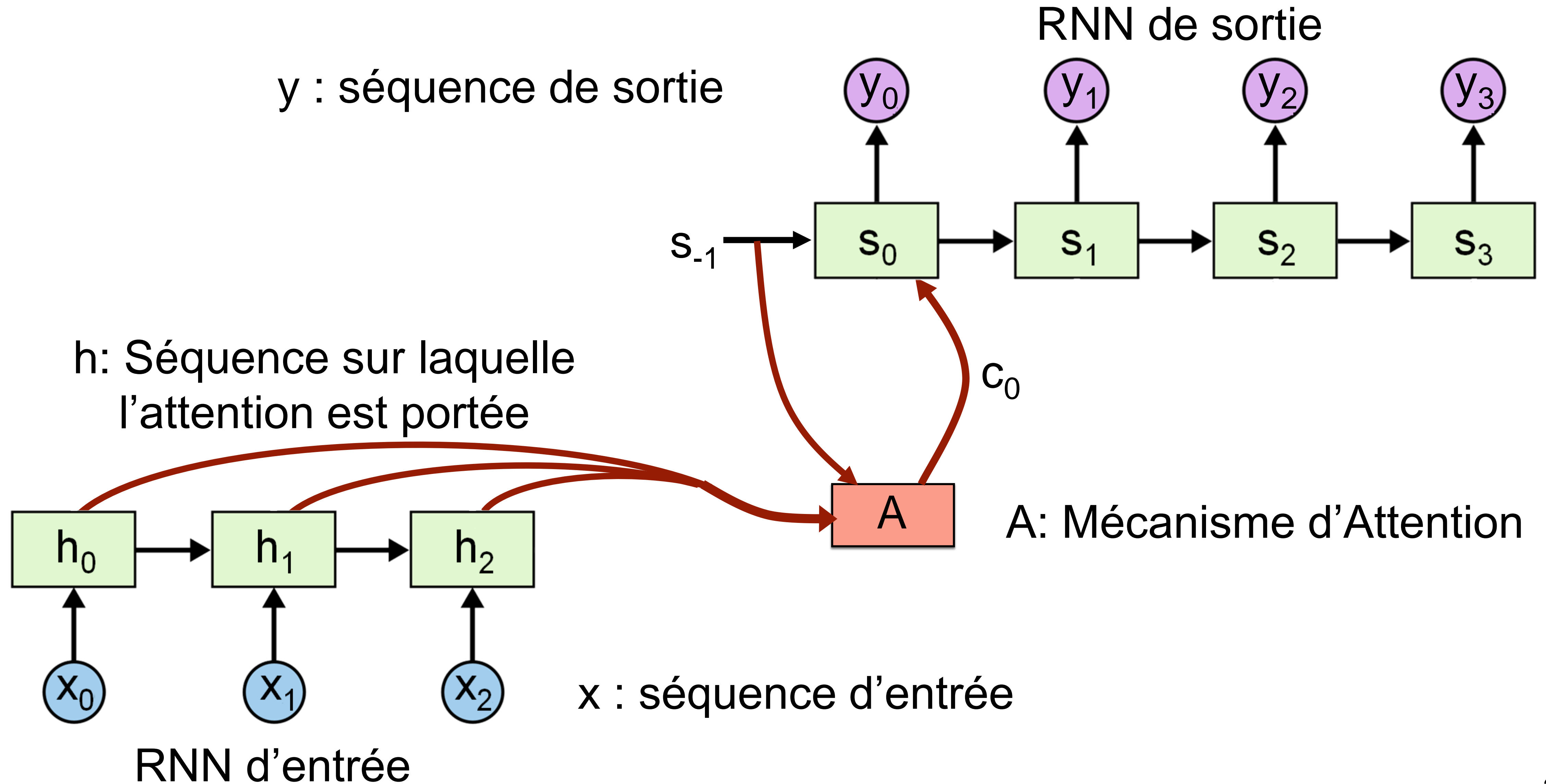


- La version la plus simple de P effectue les produits scalaires $e_i = \langle h_{t-1}, x_i \rangle$ puis les normalise à l'aide d'un softmax.
- P peut aussi être un réseau de neurones (et même profond!).



Mécanismes d'attention

Avec RNN



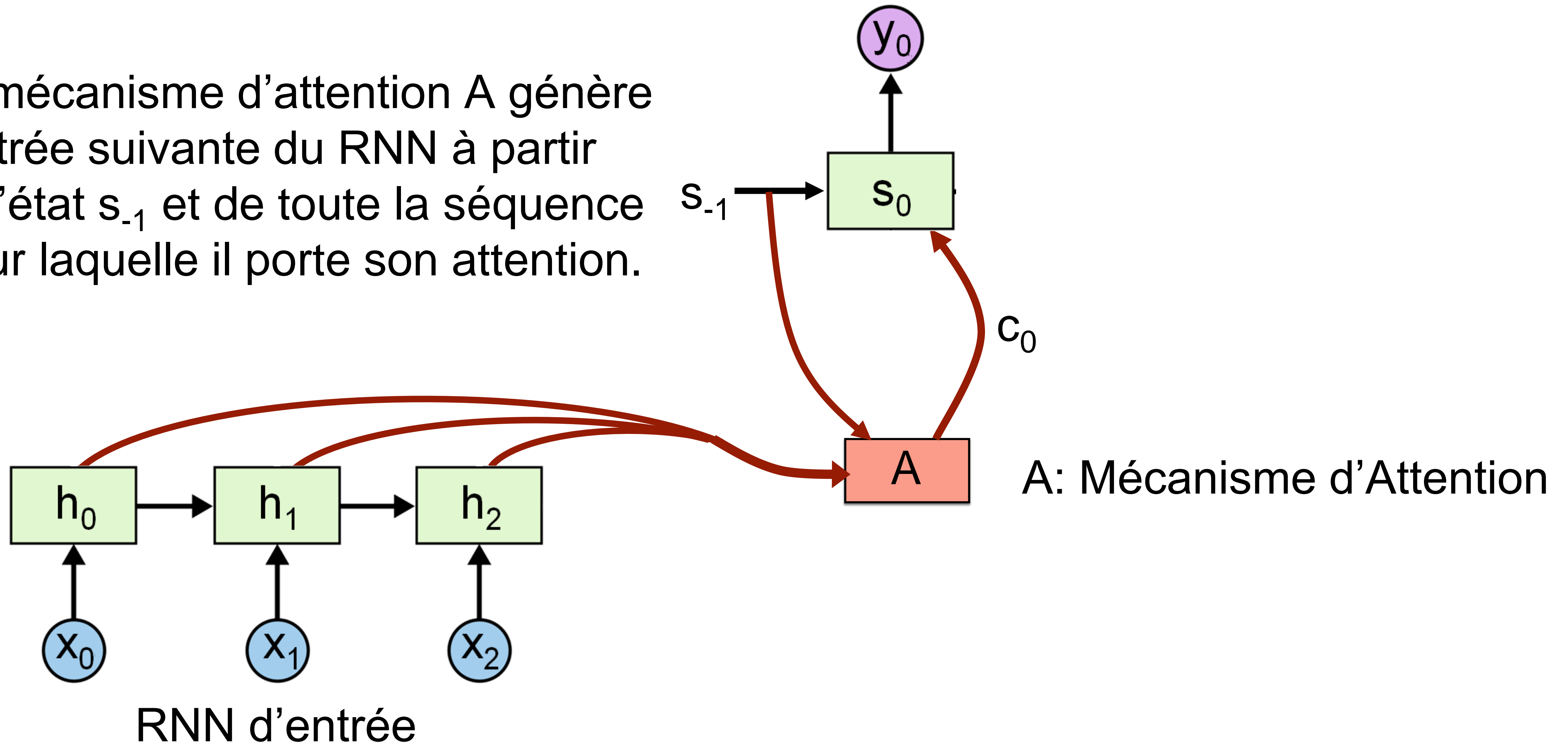
Mécanismes d'attention

Premier temps



RNN de sortie

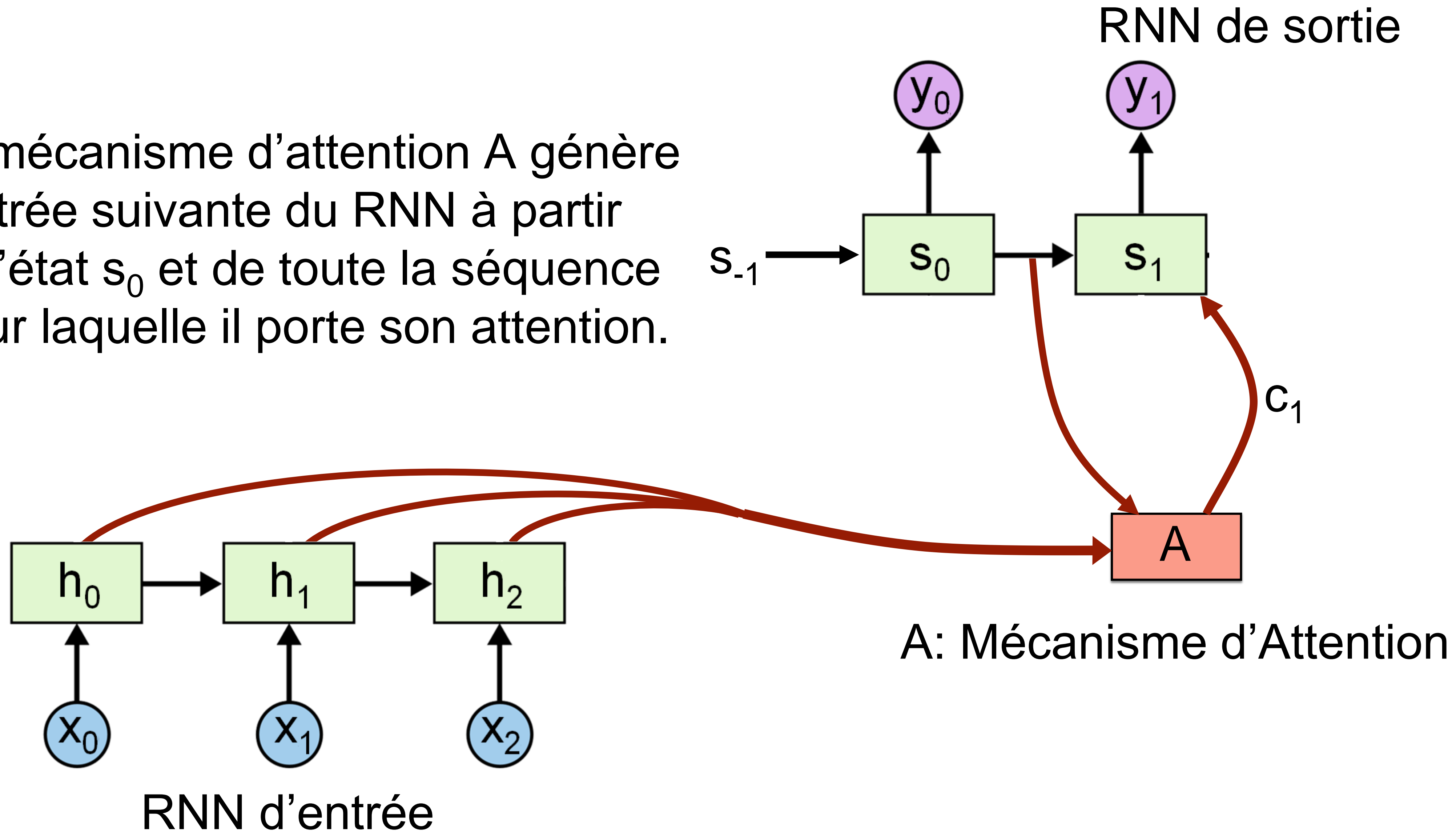
Le mécanisme d'attention A génère l'entrée suivante du RNN à partir de l'état s_{-1} et de toute la séquence h sur laquelle il porte son attention.



Mécanismes d'attention

Deuxième temps

Le mécanisme d'attention A génère l'entrée suivante du RNN à partir de l'état s_0 et de toute la séquence h sur laquelle il porte son attention.

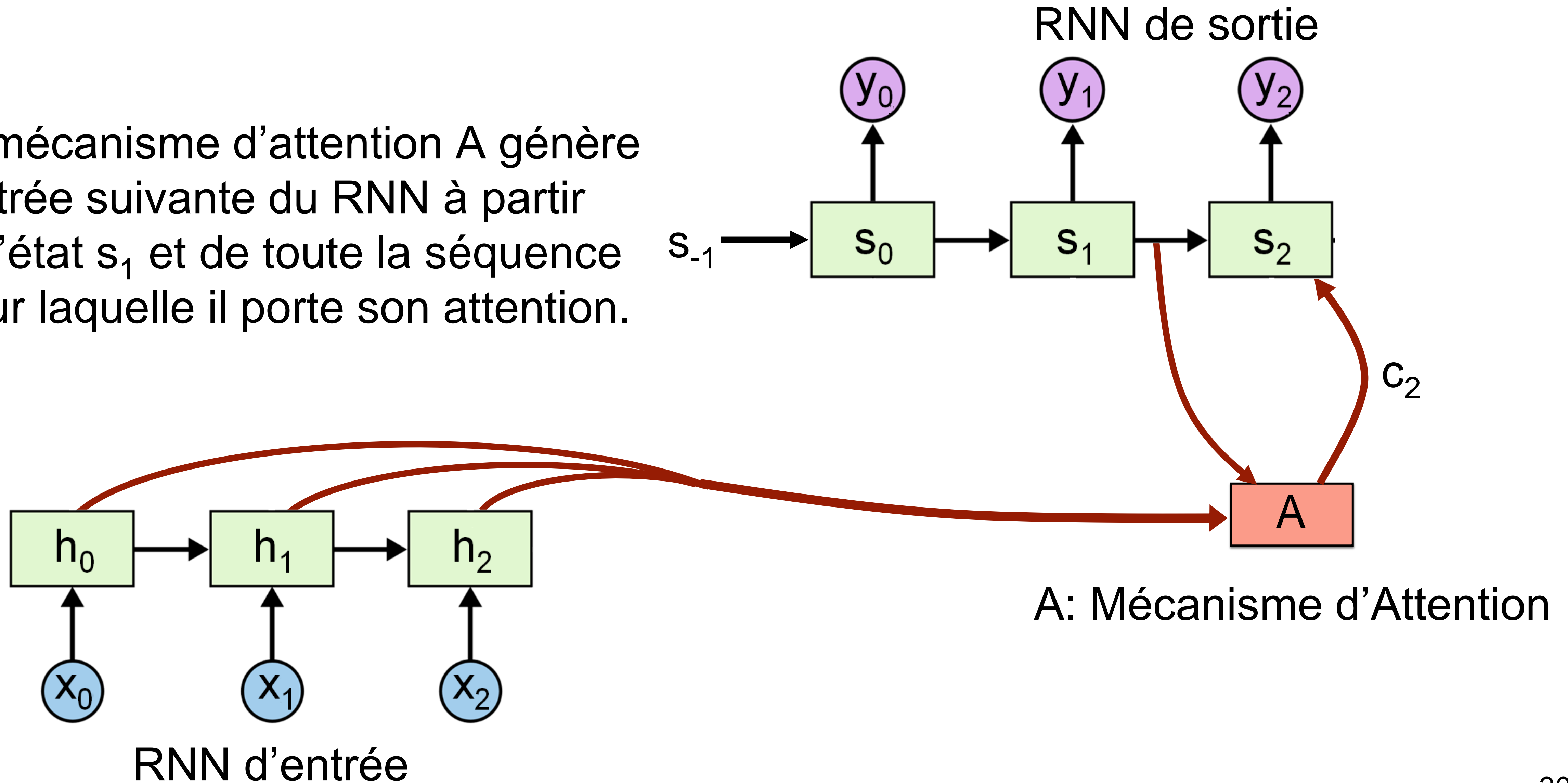


Mécanismes d'attention

Troisième temps



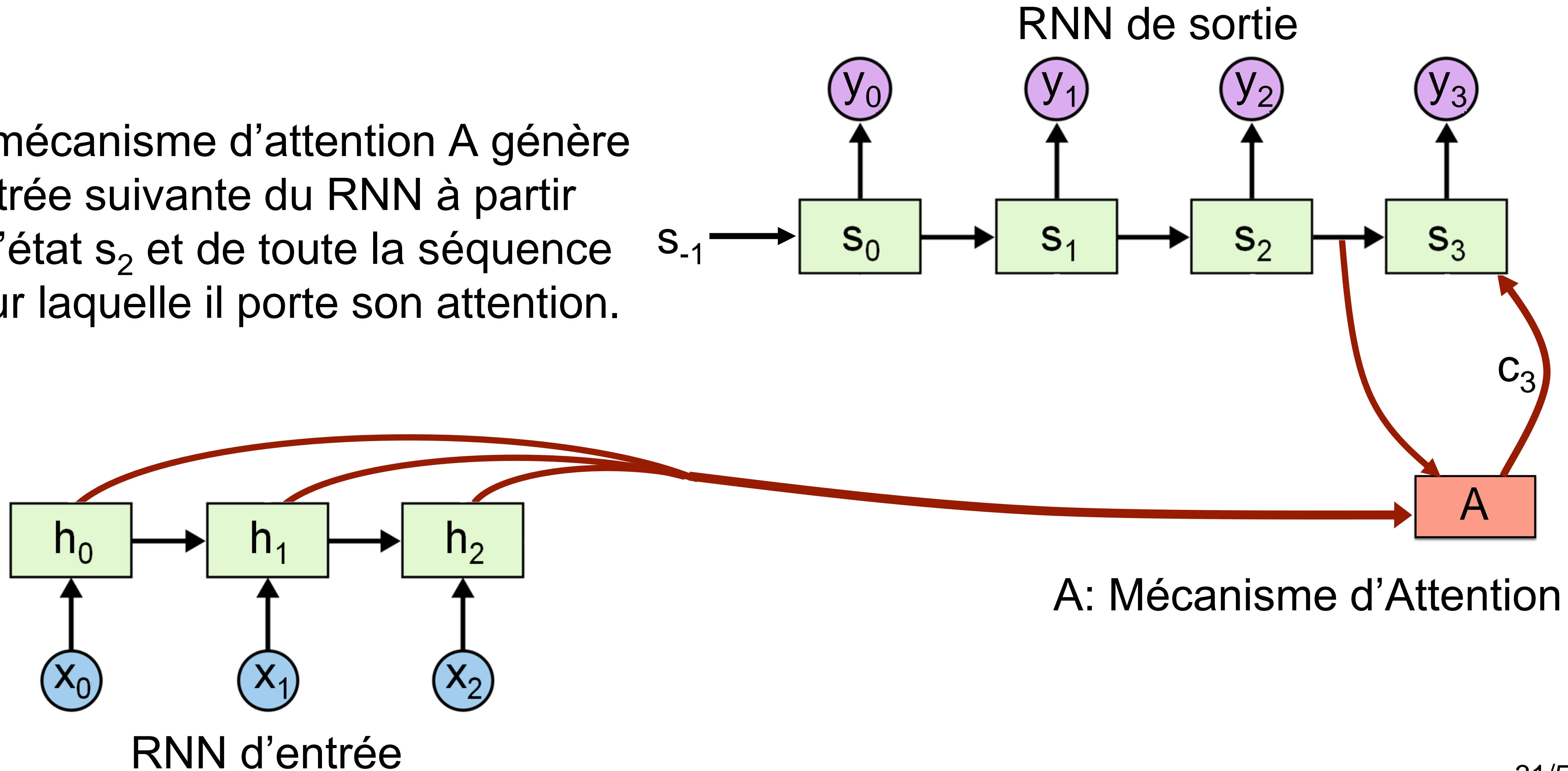
Le mécanisme d'attention A génère l'entrée suivante du RNN à partir de l'état s_1 et de toute la séquence h sur laquelle il porte son attention.



Mécanismes d'attention

Quatrième temps

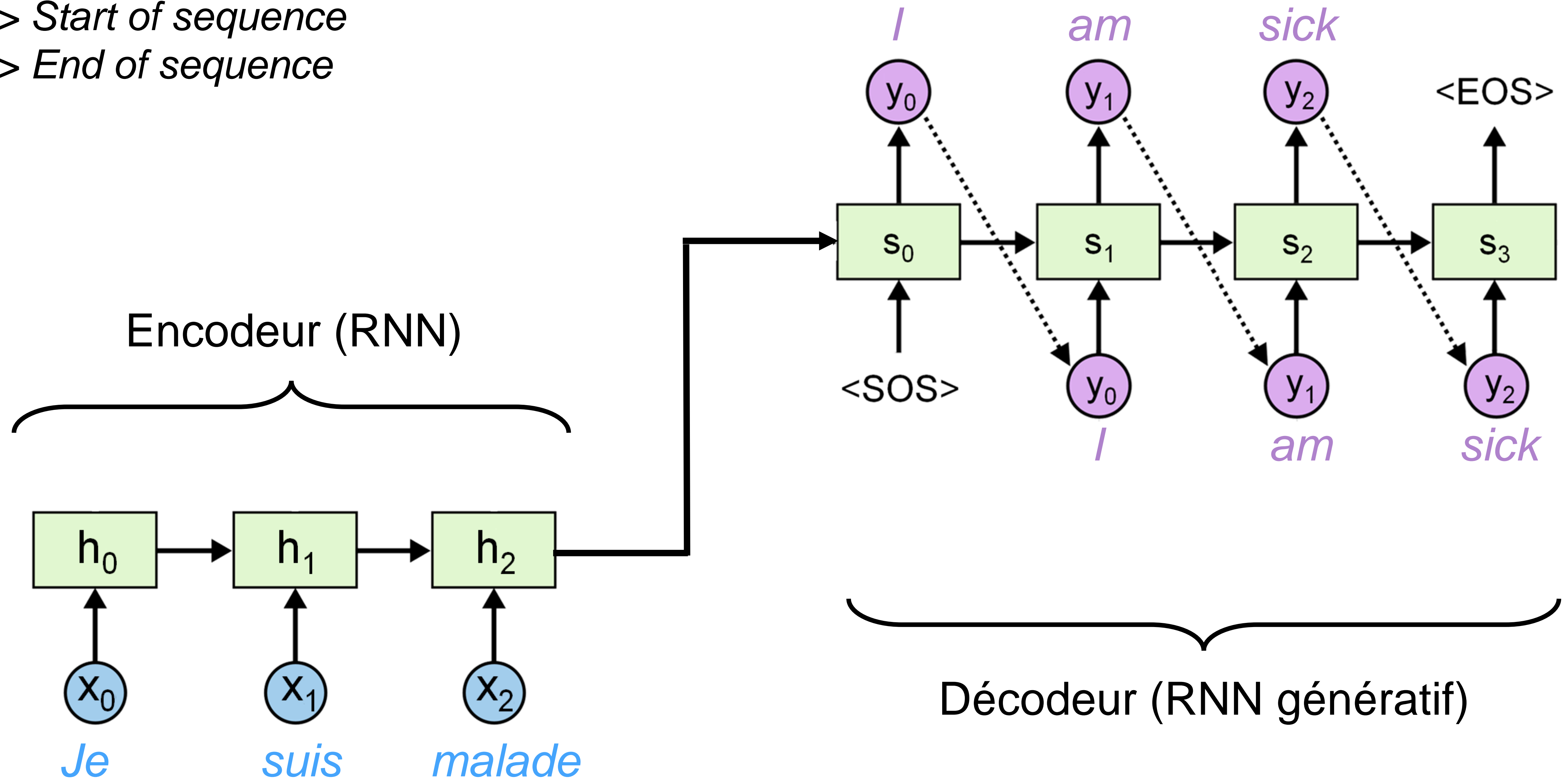
Le mécanisme d'attention A génère l'entrée suivante du RNN à partir de l'état s_2 et de toute la séquence h sur laquelle il porte son attention.



Exemple: « seq2seq »

Rappel

<SOS> Start of sequence
<EOS> End of sequence

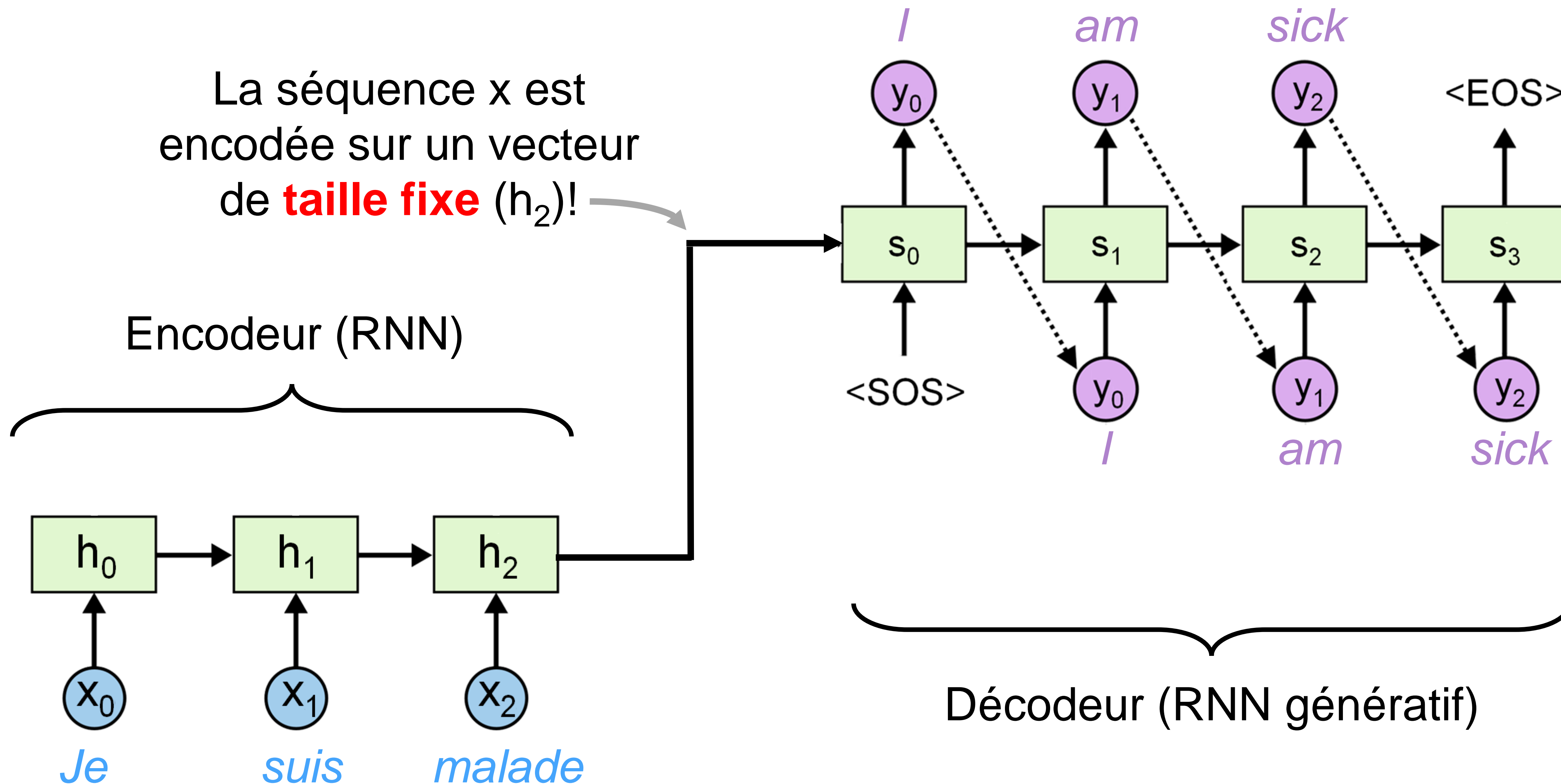


Exemple: « seq2seq »

Taille fixe!



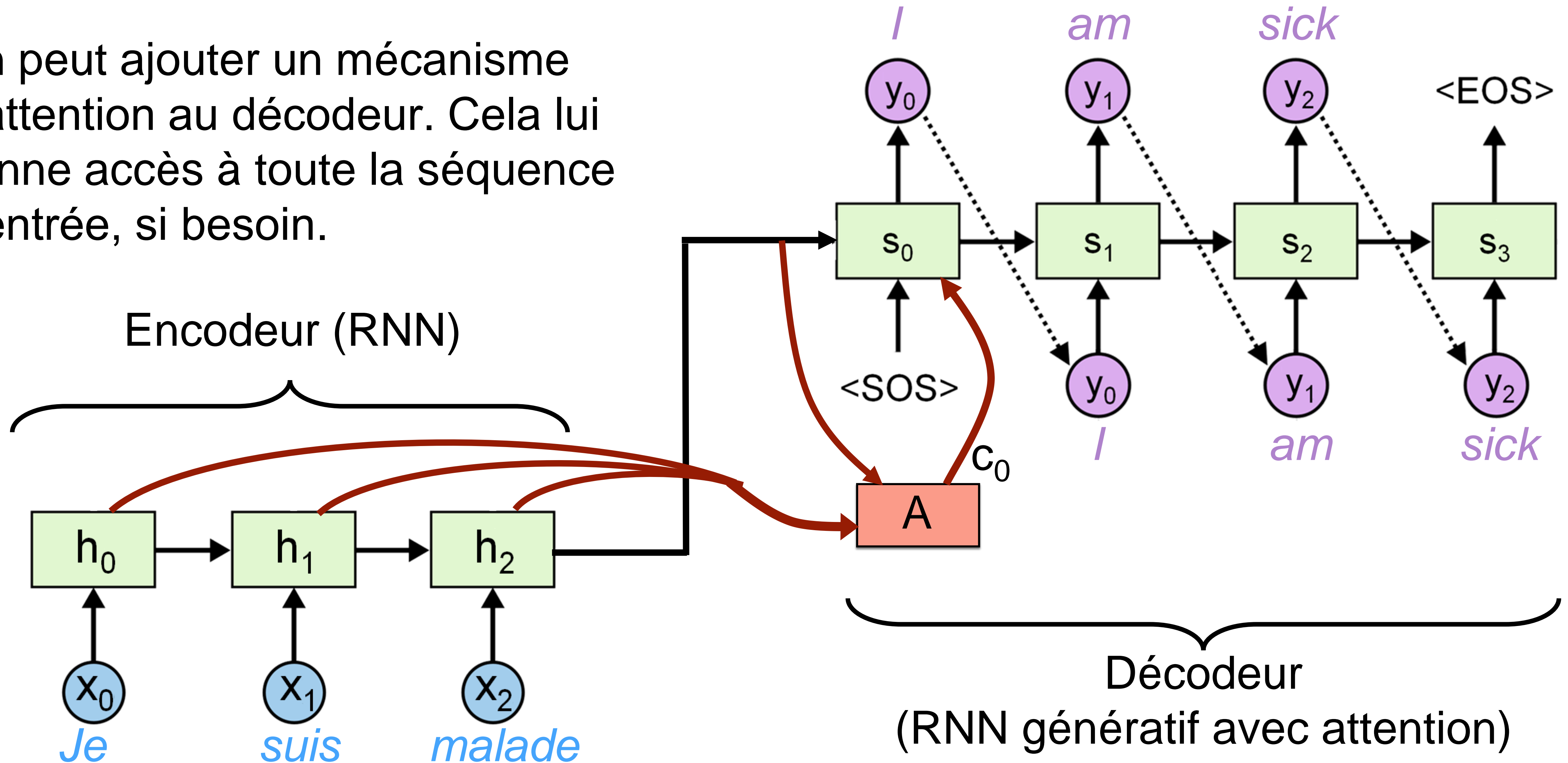
La séquence x est encodée sur un vecteur de **taille fixe** (h_2)!



Exemple: « seq2seq » + attention

Ajout du mécanisme d'attention

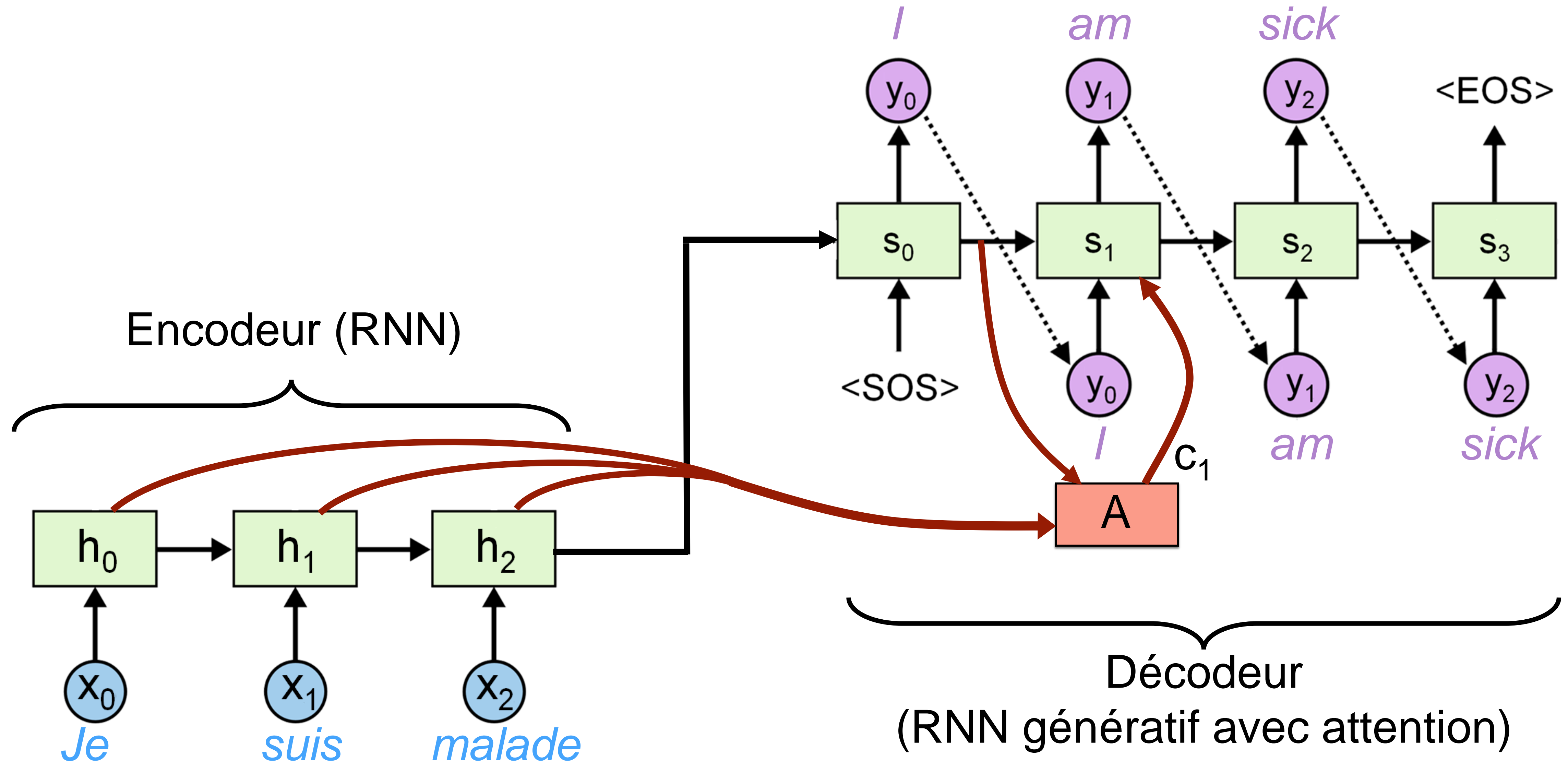
On peut ajouter un mécanisme d'attention au décodeur. Cela lui donne accès à toute la séquence d'entrée, si besoin.



Exemple: « seq2seq » + attention



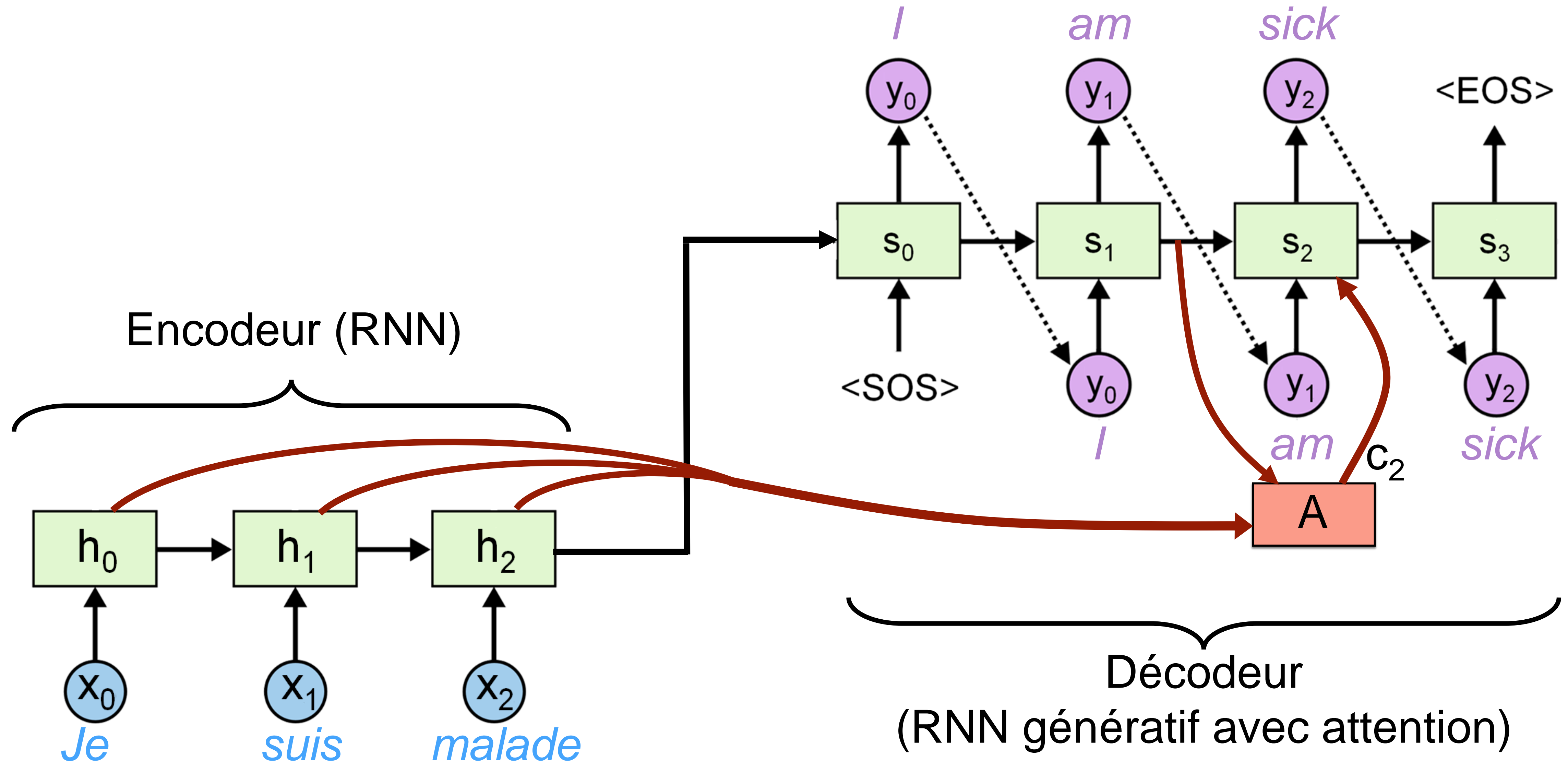
Deuxième temps



Exemple: « seq2seq » + attention



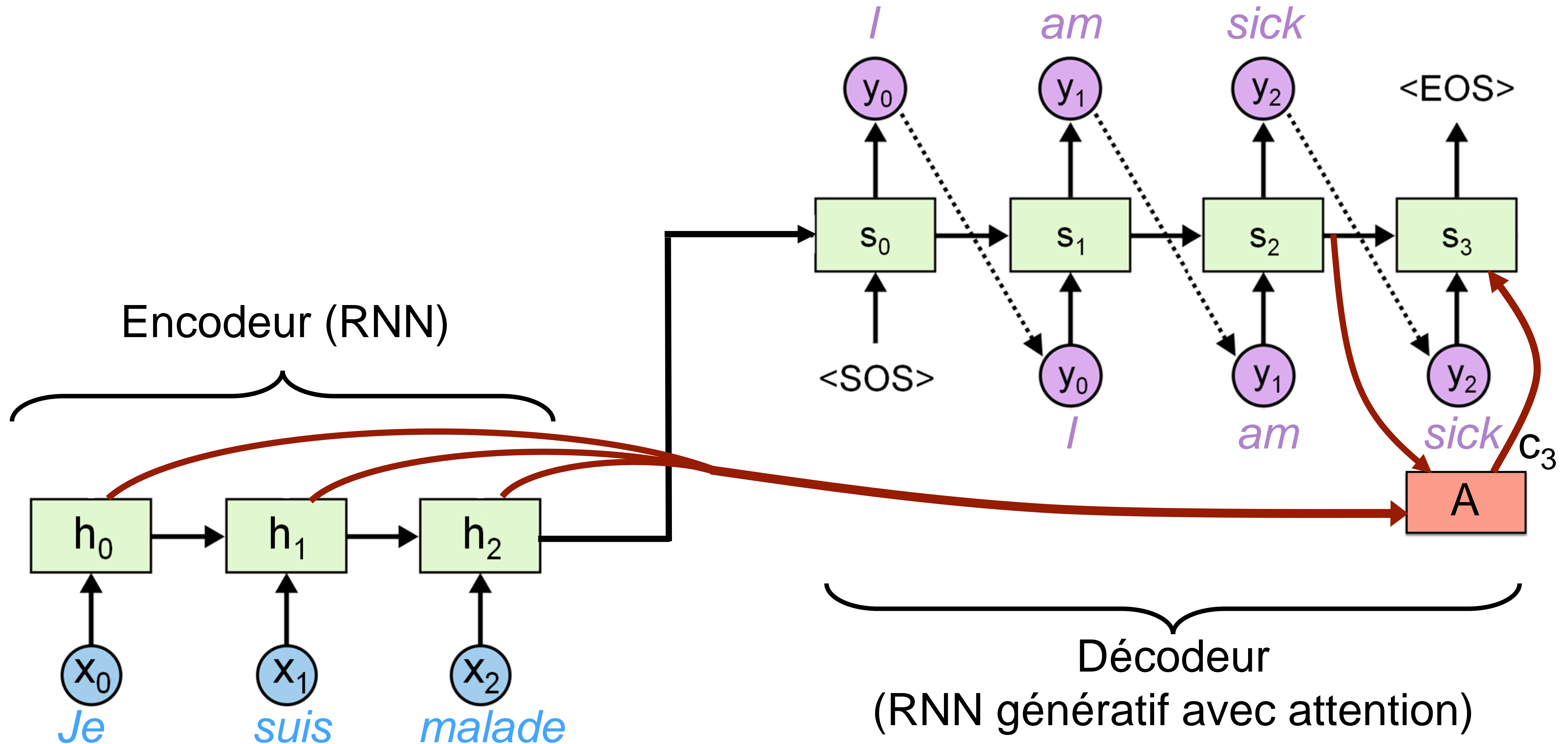
Troisième temps



Exemple: « seq2seq » + attention



Quatrième temps



Exemple: « seq2seq » + attention

Résultats comparatifs

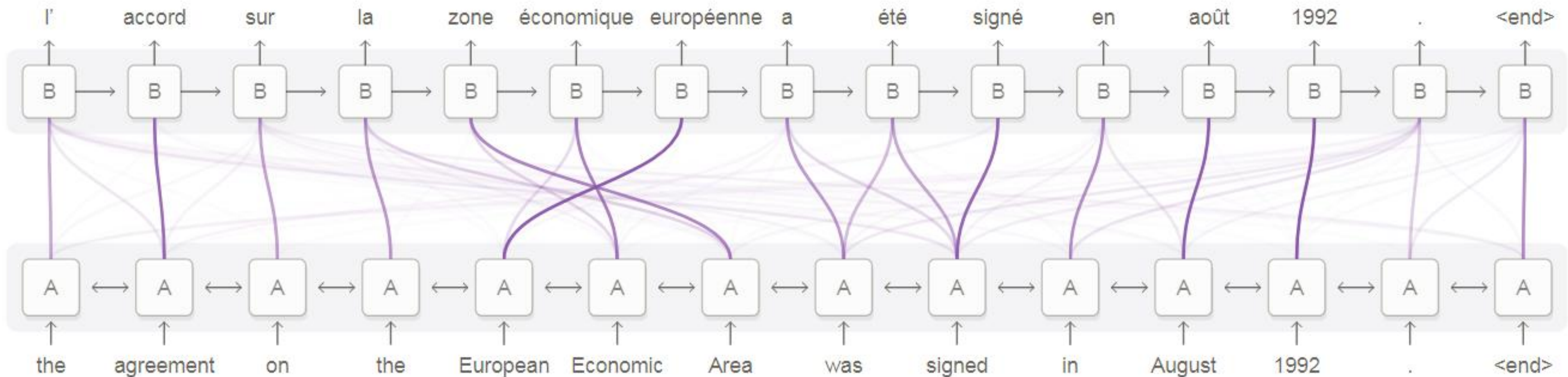


Modèle	Taille de x max.	BLEU score
Seq2seq	30	13.9
Seq2seq + attention	30	21.5
Seq2seq	50	17.8
Seq2seq + attention	50	26.7

- Le BLEU score (*BiLingual Evaluation Understudy*) mesure la qualité de la traduction (le plus grand le mieux)
- L'attention améliore grandement la qualité des traductions.

Exemple: « seq2seq » + attention

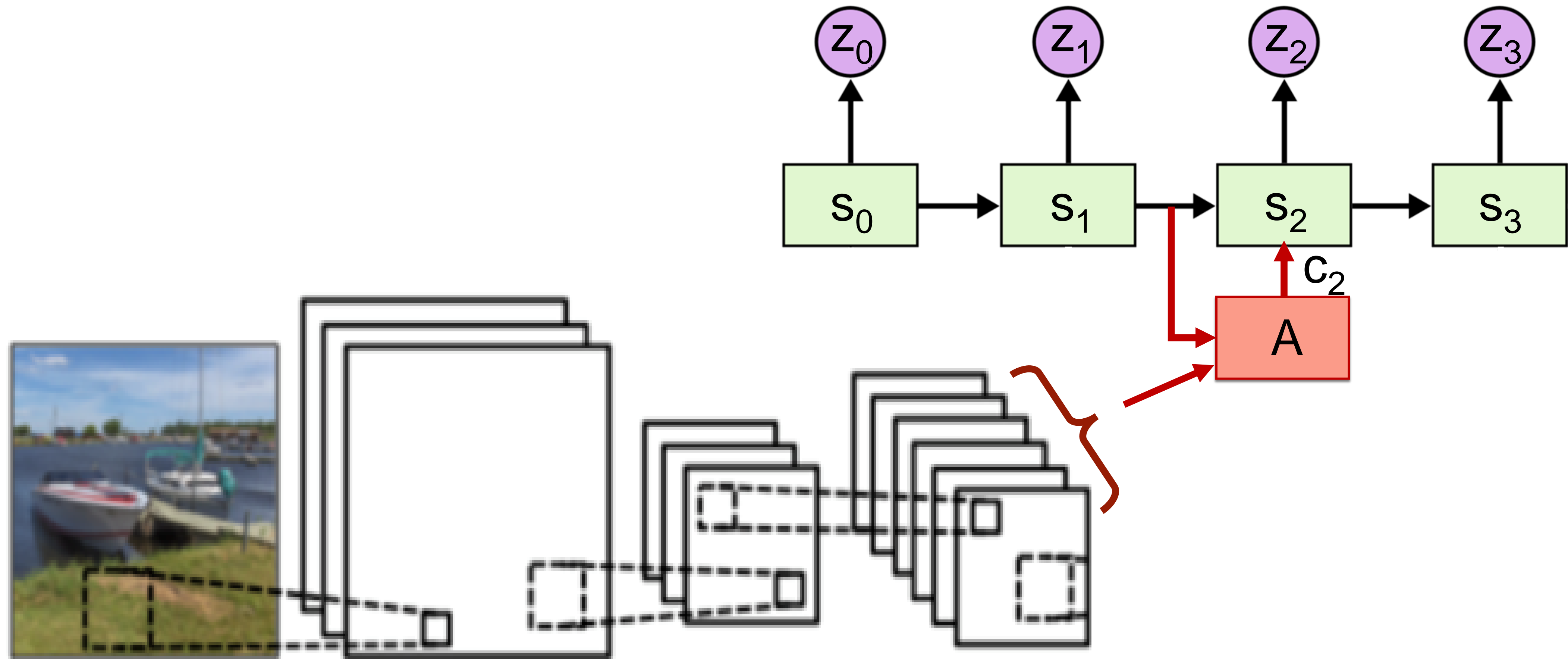
Visualisation de l'attention



Les traits mauves montrent où le RNN B (décodeur) porte son attention dans la séquence générée par le RNN A (encodeur).

Mécanismes d'attention

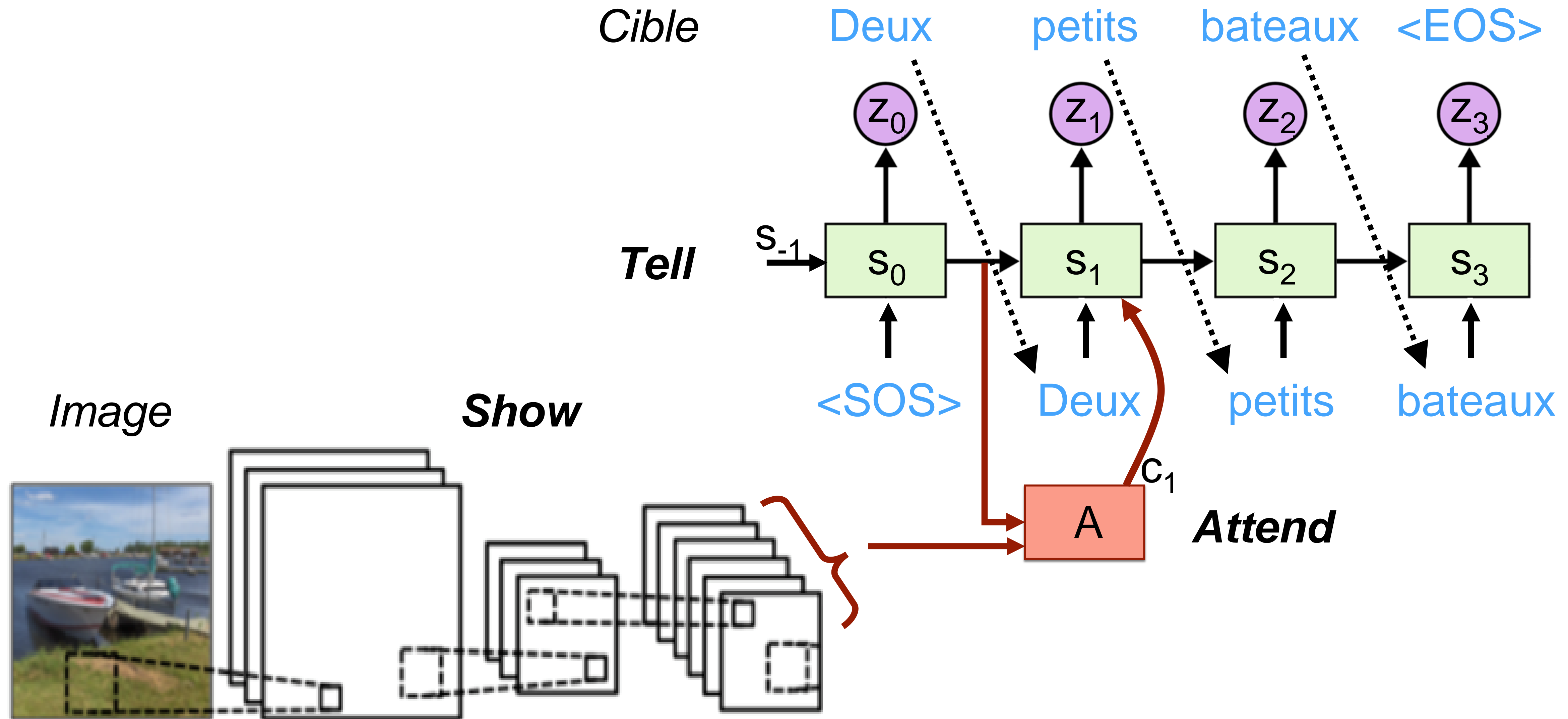
Sur une image



Un mécanisme d'attention peut aussi être utilisé sur une image!

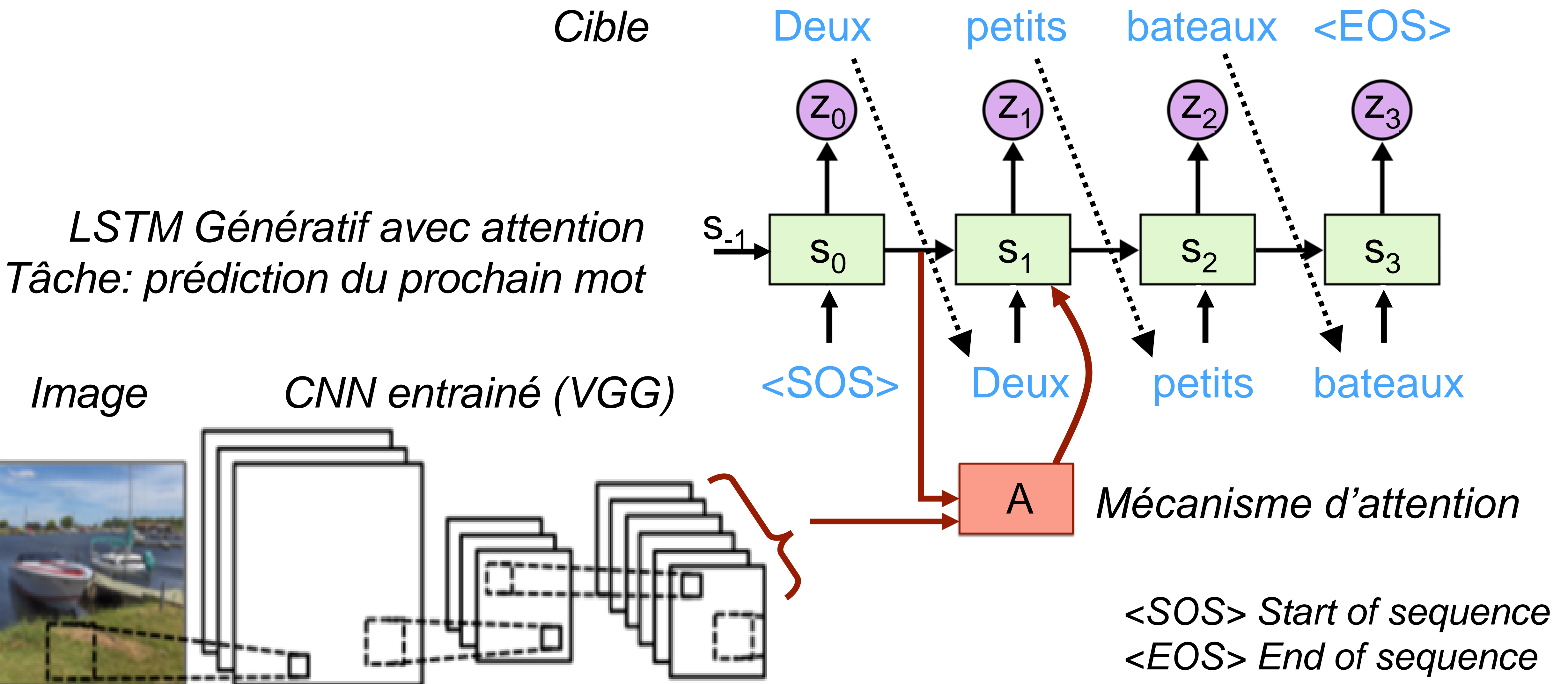
Génération de légende

Modèle « Show Attend and Tell »



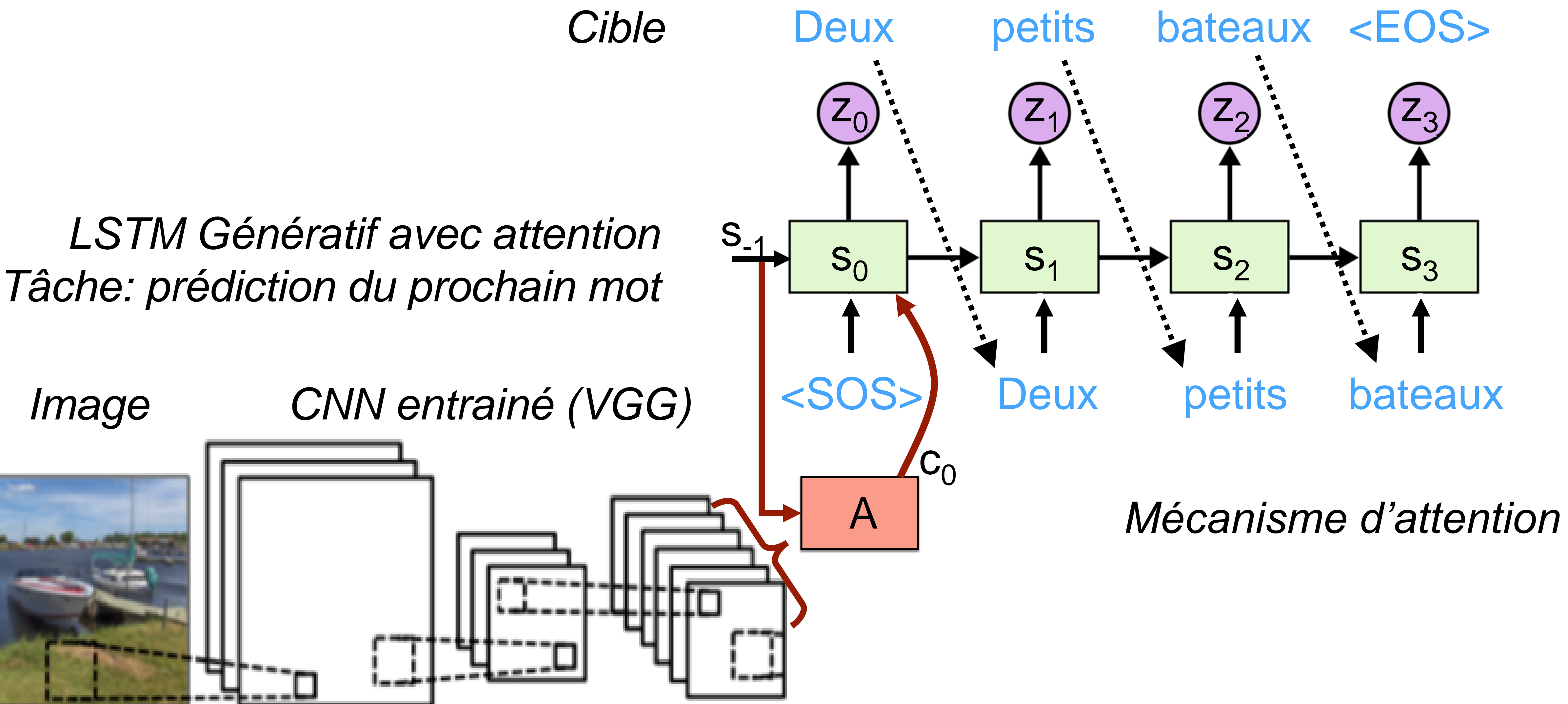
Génération de légende

En détails



Génération de légende

Entraînement: Génération du premier mot (à partir de $\langle \text{SOS} \rangle$)



Génération de légende

Entraînement: Génération du deuxième (à partir de « Deux »)

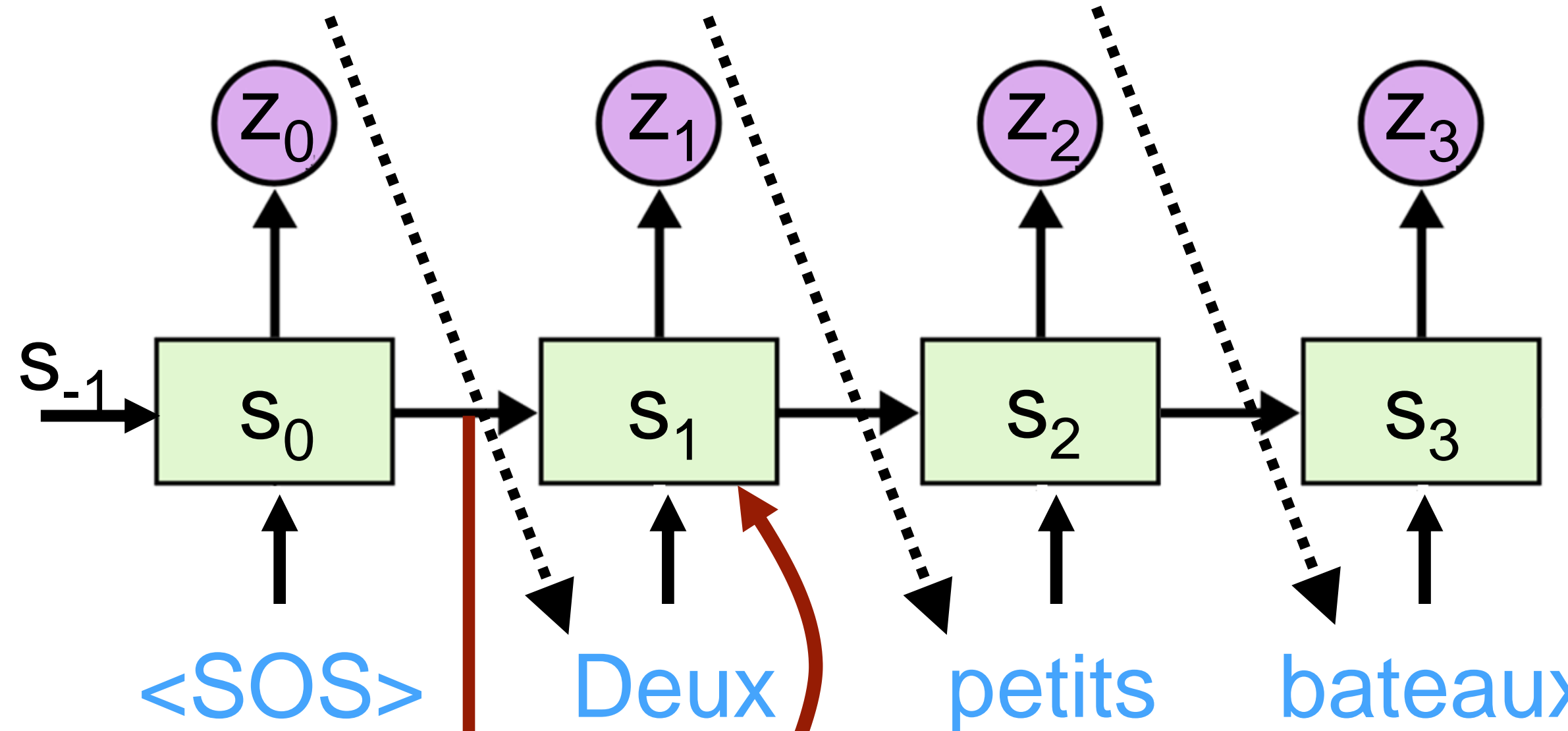
Cible

Deux

petits

bateaux

<EOS>

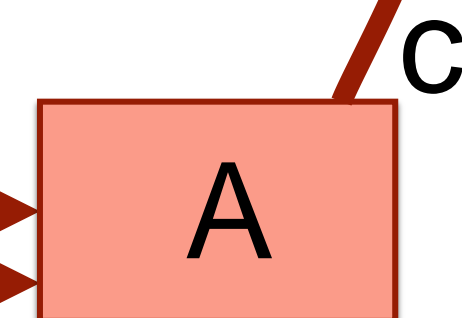
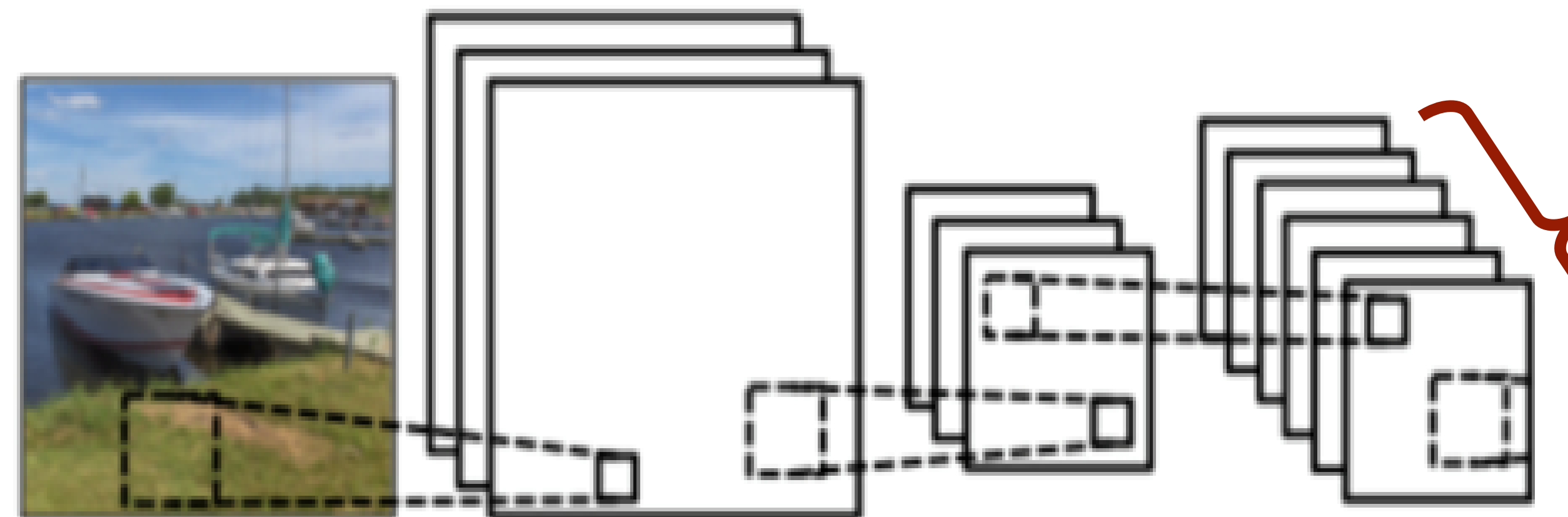


LSTM Génératif avec attention

Tâche: prédiction du prochain mot

Image

CNN entraîné (VGG)



Mécanisme d'attention

Génération de légende

Entraînement: Génération du troisième mot (à partir de « petits »)

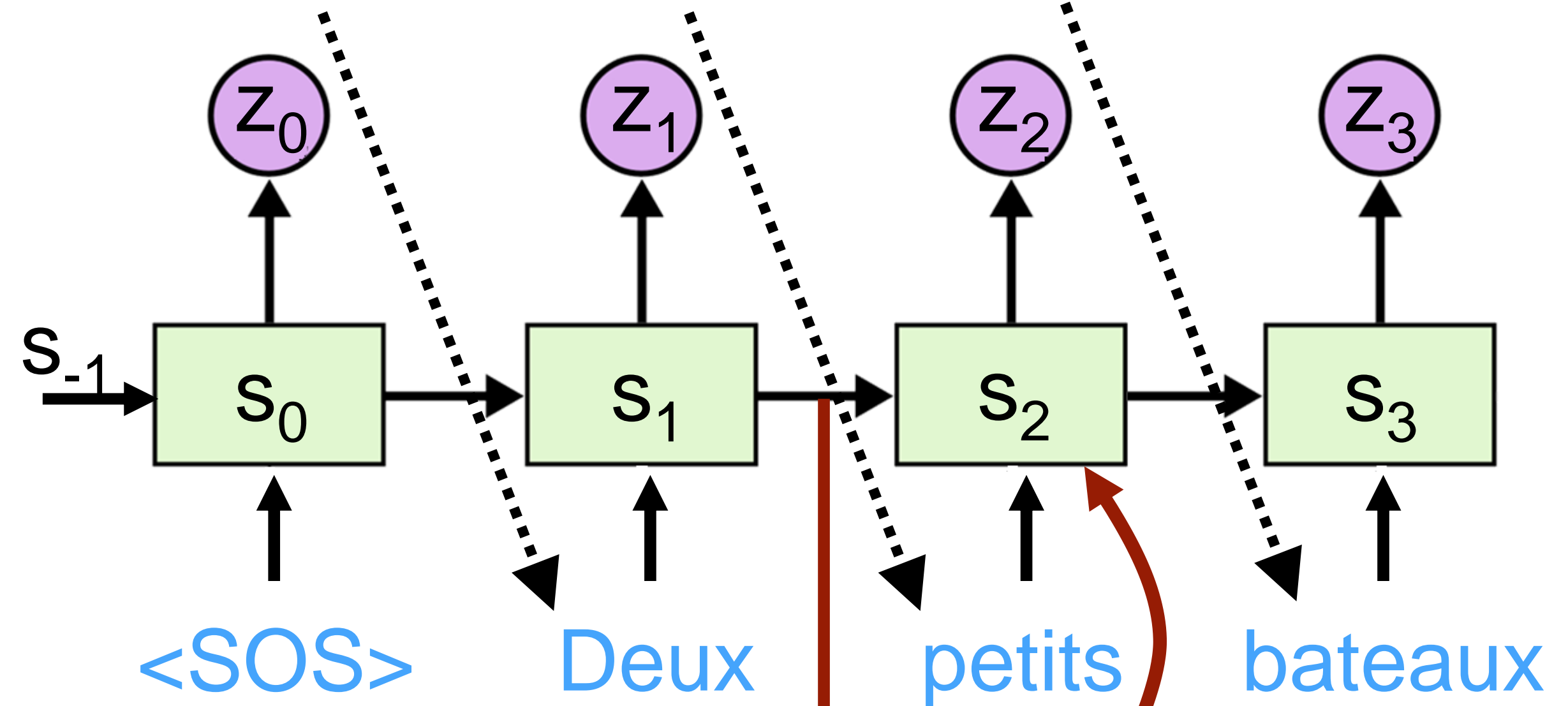
Cible

Deux

petits

bateaux

<EOS>



LSTM Génératif avec attention

Tâche: prédiction du prochain mot

Image

CNN entraîné (VGG)



Mécanisme d'attention

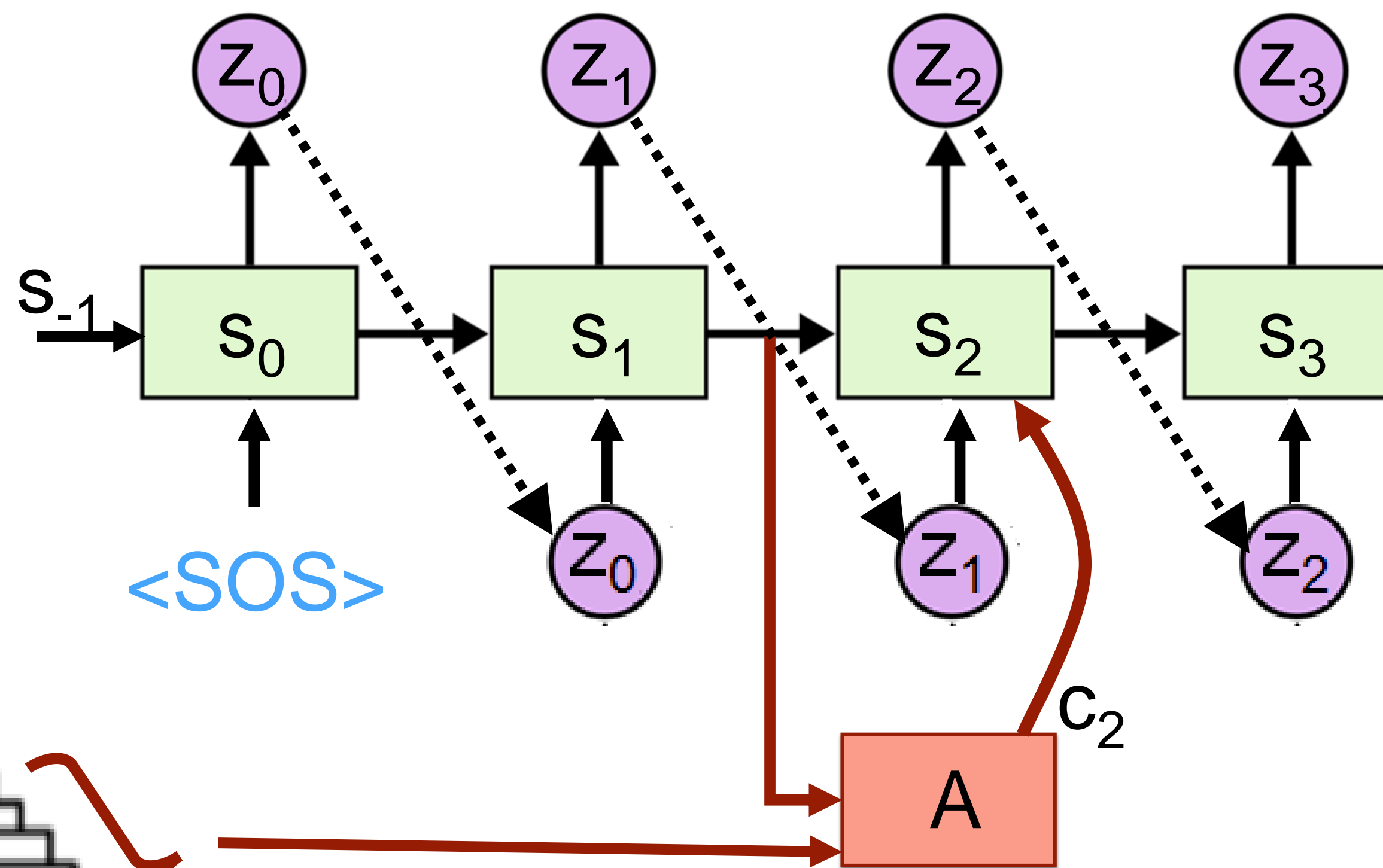
Génération de légende

Test: Utilisation de la séquence z

Mot généré le plus probable:

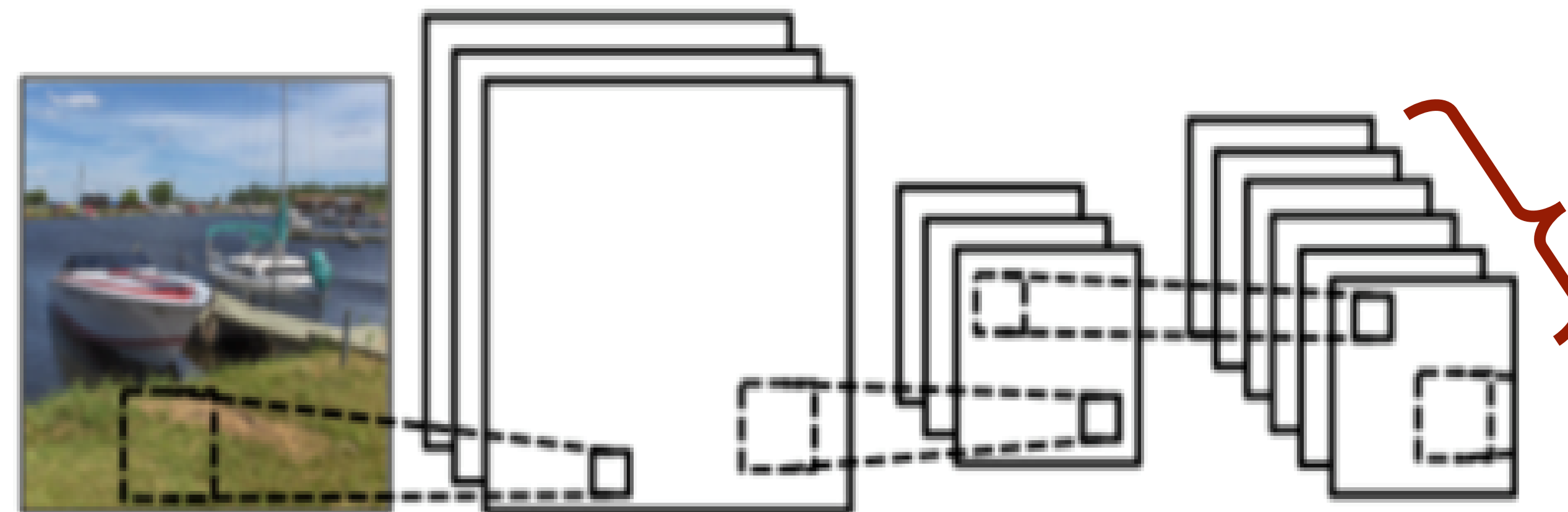
Un bateau blanc <EOS>

*LSTM Génératif
Avec attention*



Image

CNN entraîné (VGG)



Mécanisme d'attention

Génération de légende

Résultats

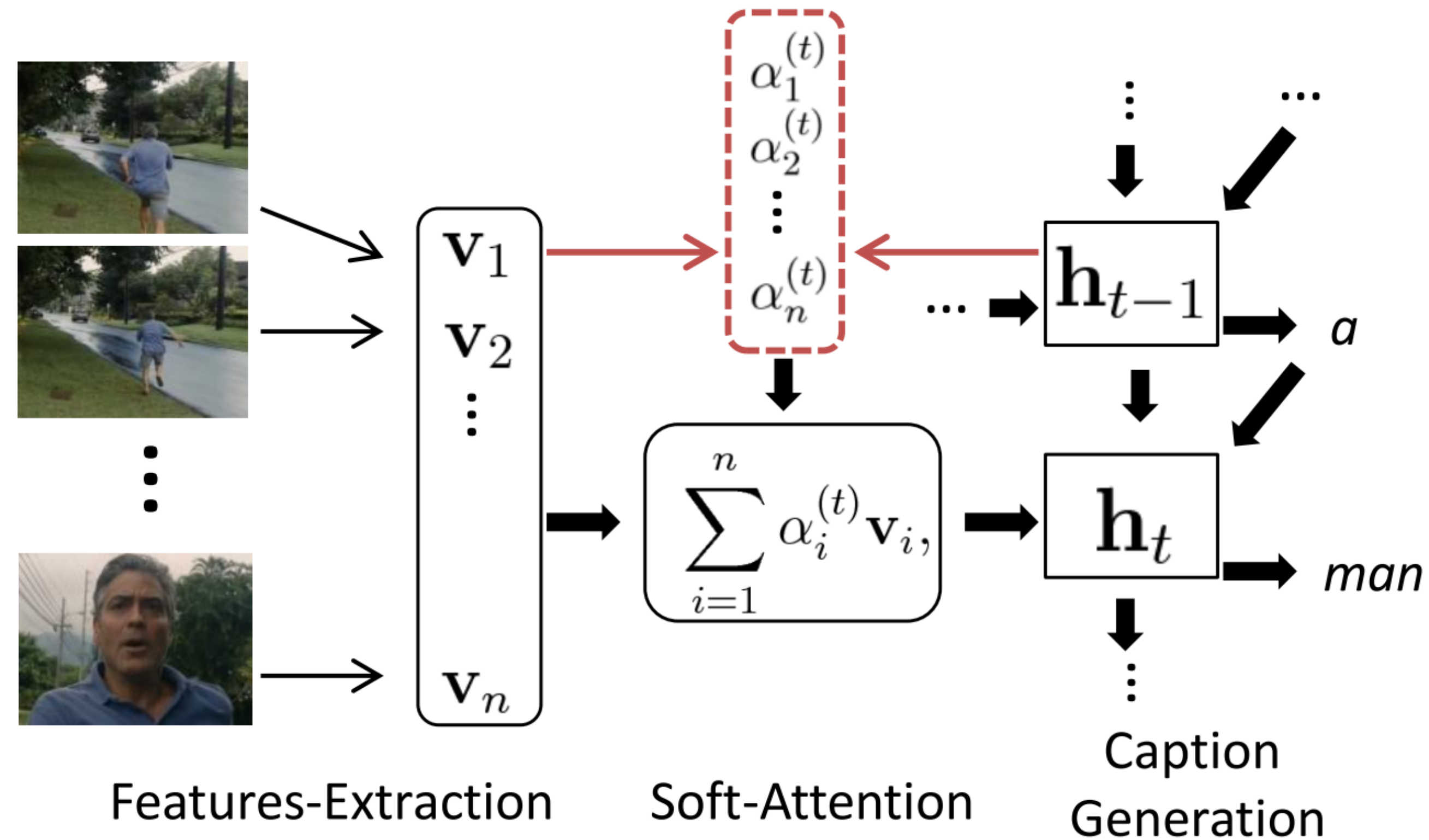


Génération de légende

Pour vidéos

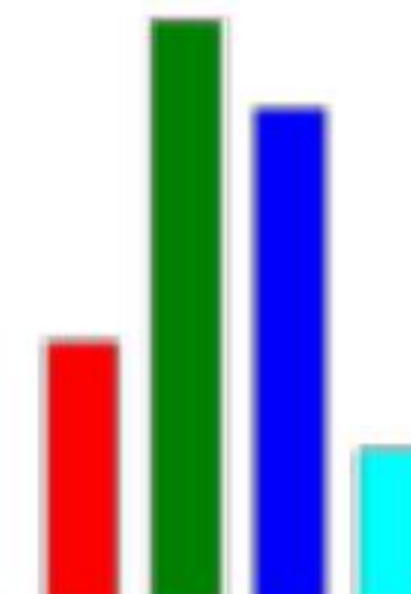
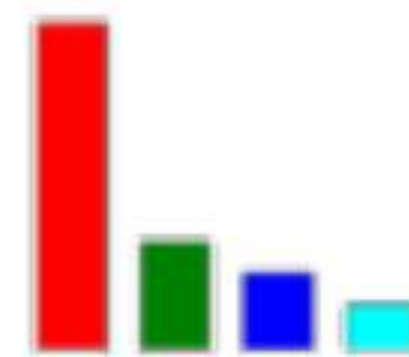
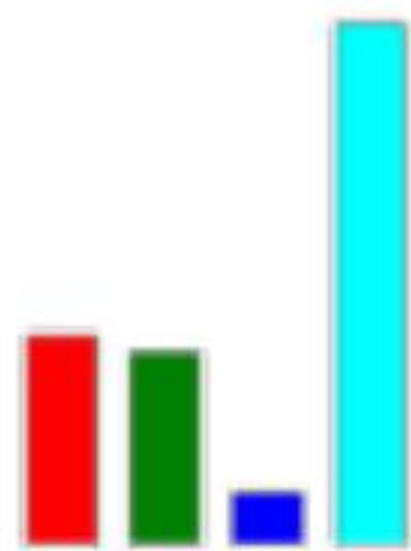


On peut aussi le faire sur des **vidéos!**



Génération de légende

Résultats



+Local+Global: **Someone** is **frying** a **fish** in a **pot**

Ref: A woman is frying food

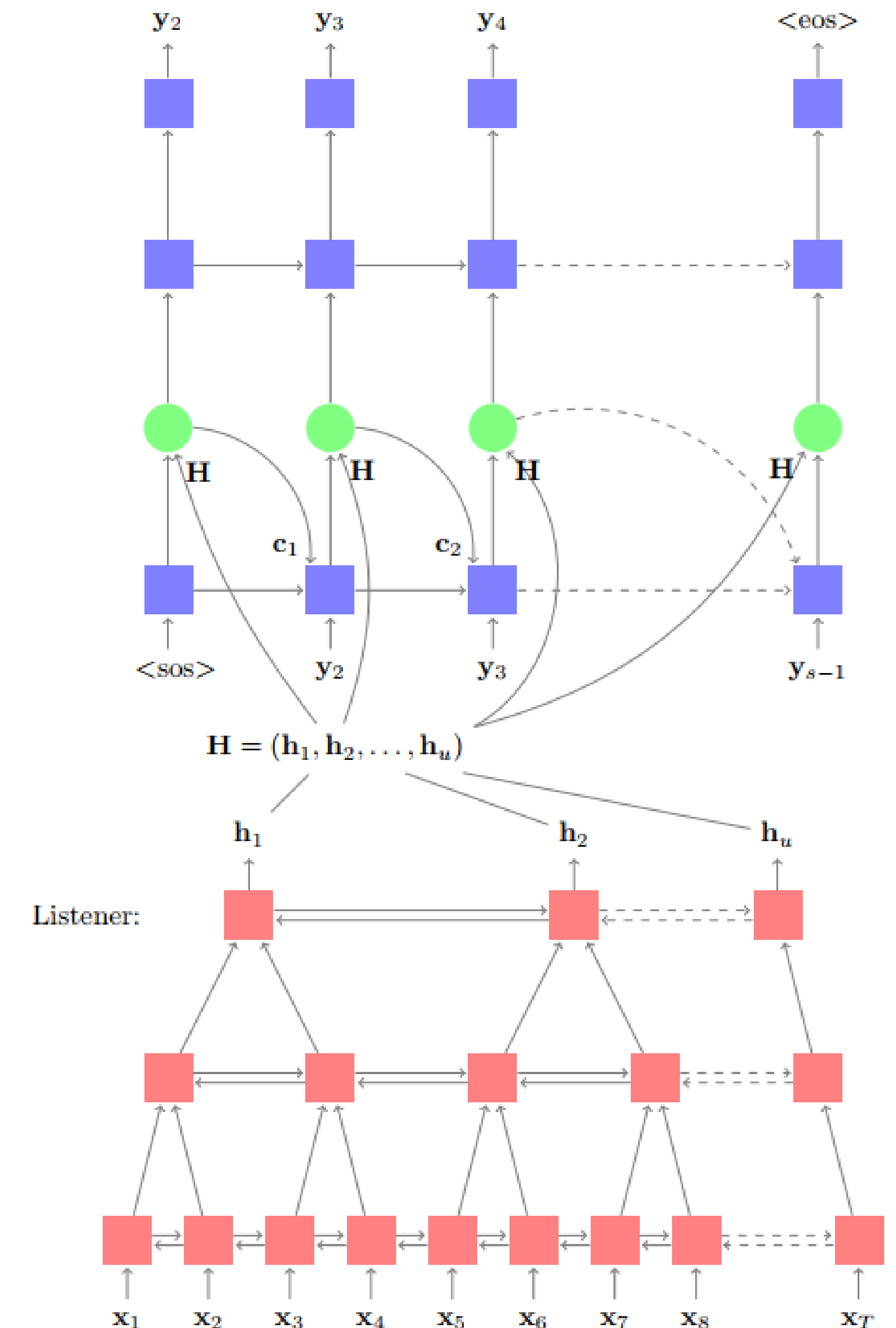
Reconnaissance de la parole



Modèle « Listen Attend and Spell »

Systeme complet de reconnaissance de la parole.

- Entrée: Spectrogramme
- Sortie: 29 Caractères (26 lettres + espace + <SOS> + <EOS>)
- 3 Parties: **Listener**, **Attender** and **Speller**



Reconnaissance de la parole

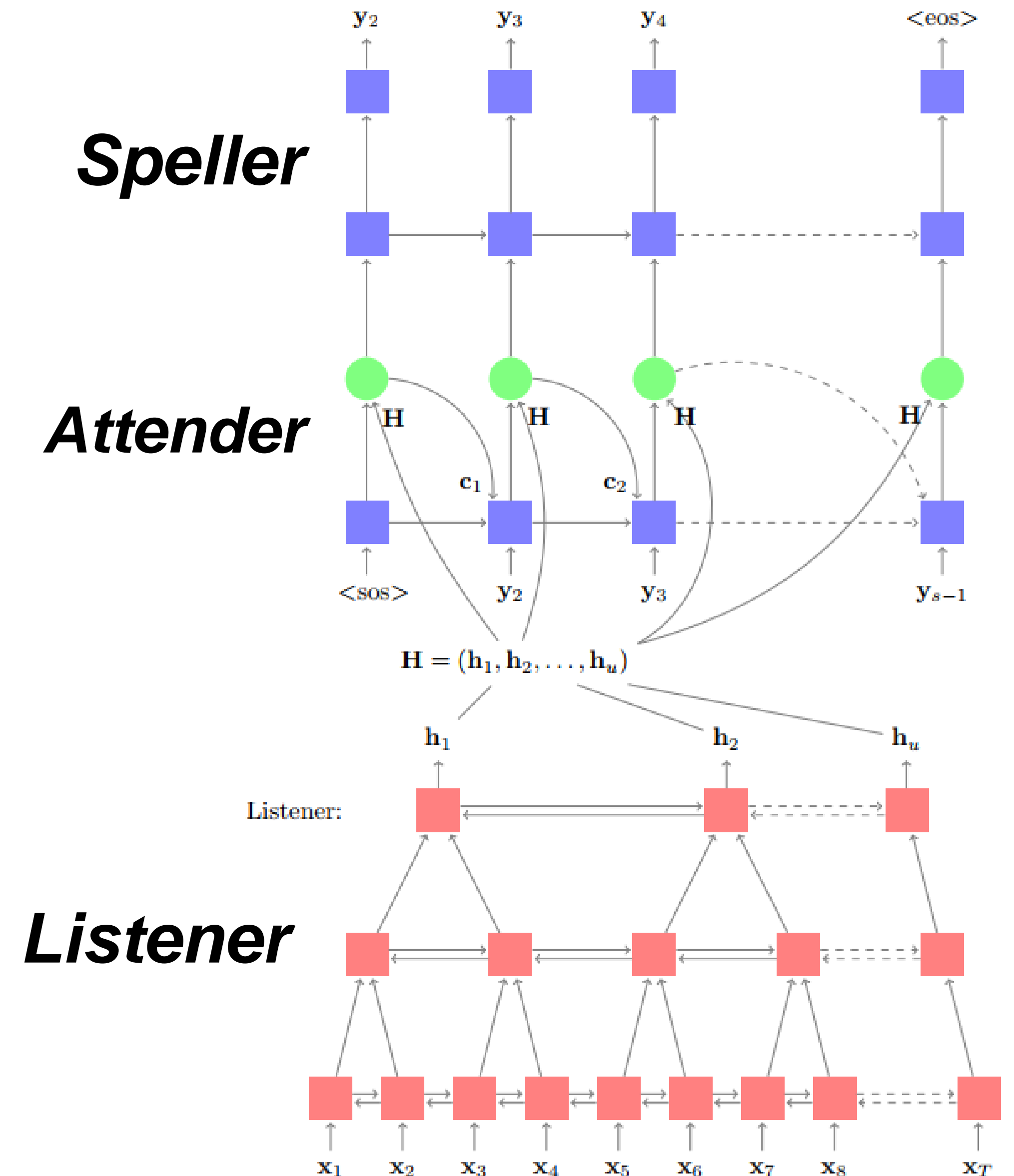


Modèle « Listen Attend and Spell »

Systeme complet de reconnaissance de la parole.

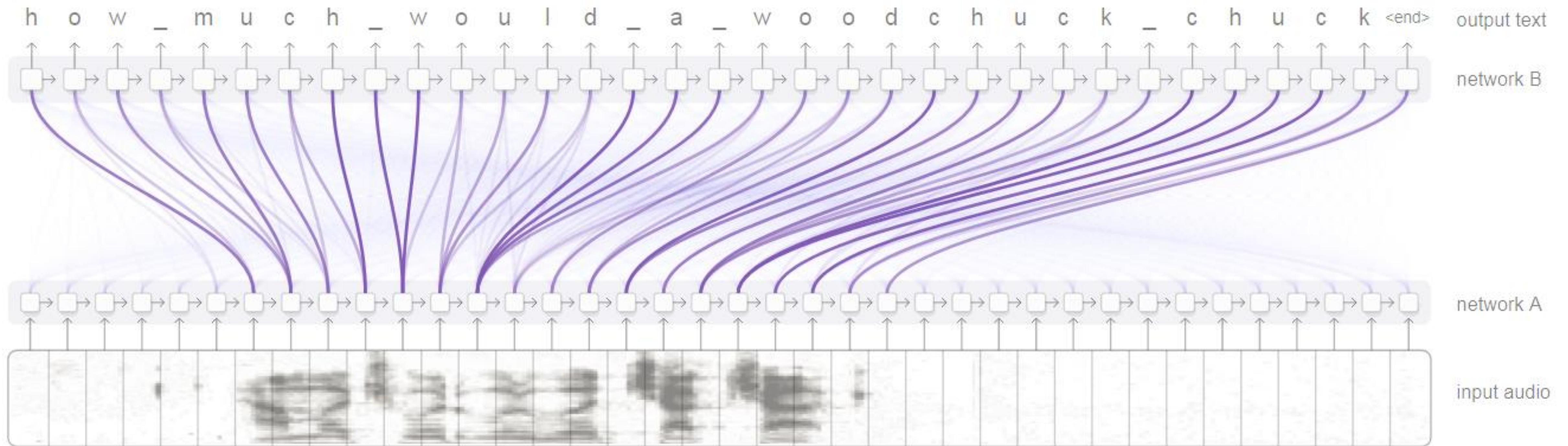
- Listener: Traite la séquence d'entrée
- Attender: Choisit où écouter dans la séquence traitée
- Speller: Génère les caractères

B-LSTMs, **LSTMs**, **Attention**

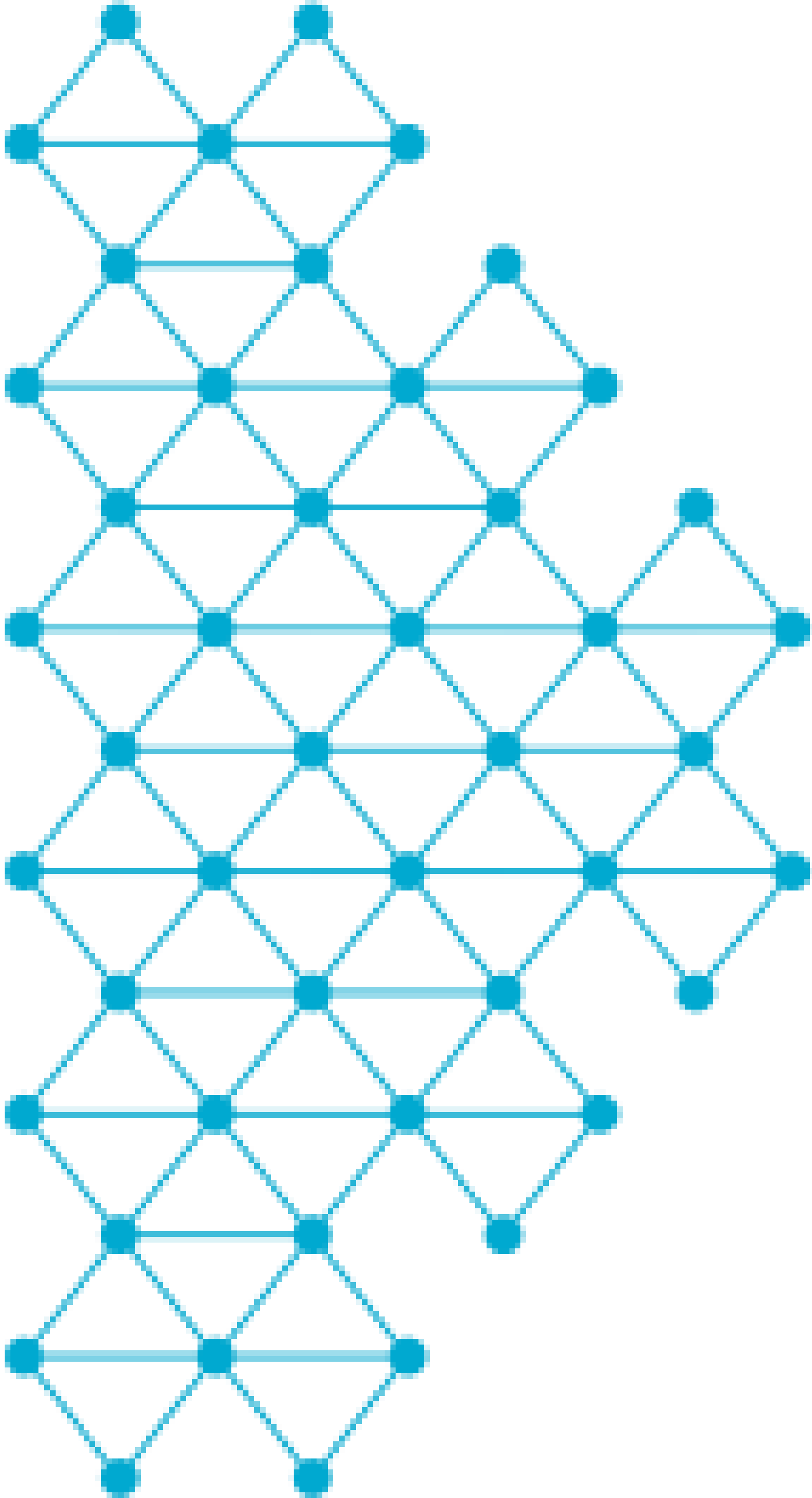


Reconnaissance de la parole

Visualisation de l'attention



Les traits mauves montrent où le RNN B (Attendre) porte son attention dans la séquence générée par le RNN A (Listener).



1. Retour sur la première partie

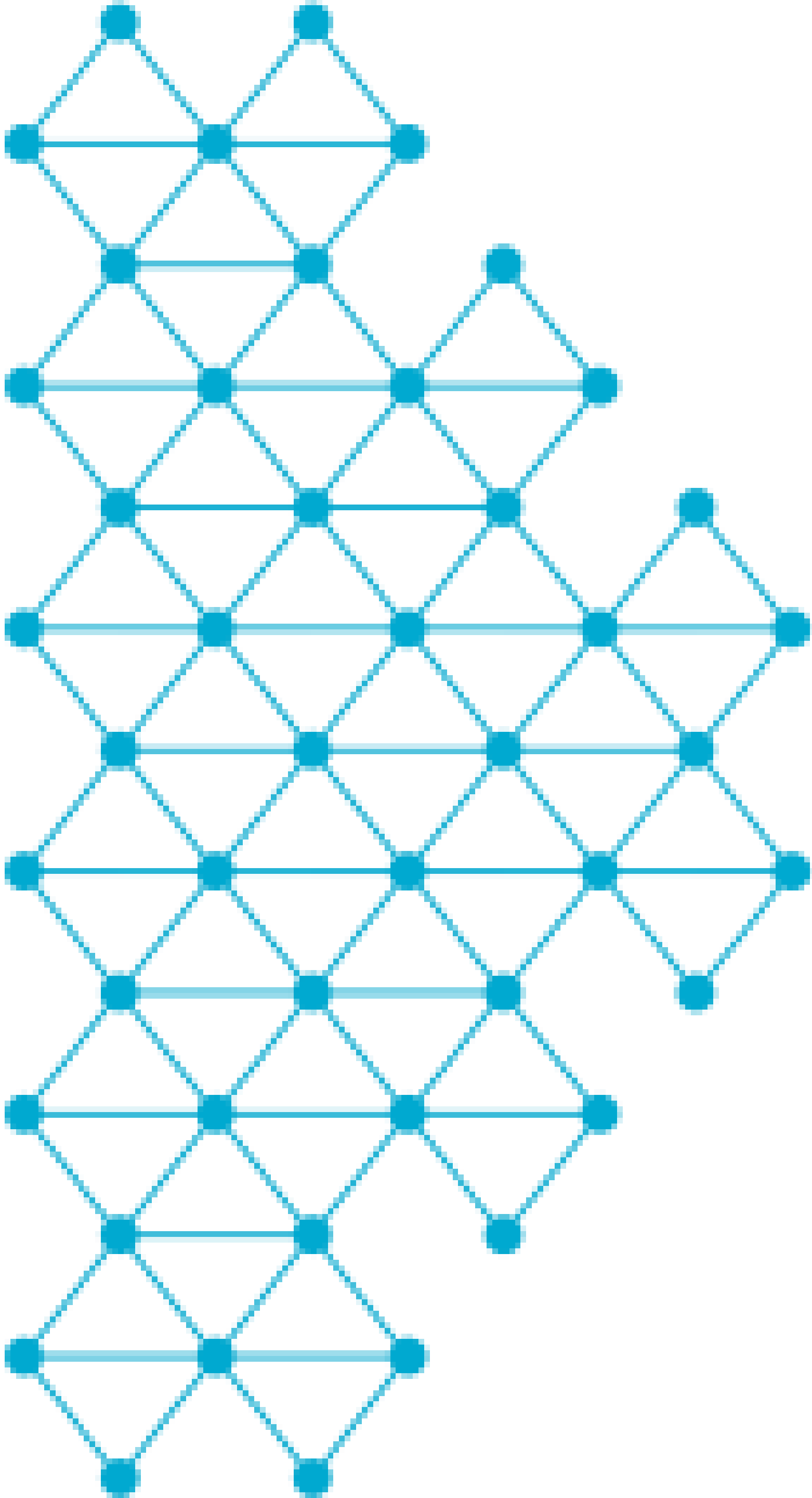
2. RNNs génératifs

3. Mécanismes d'attention

4. Bibliothèques et références

- Les RNNs sont présents dans les librairies de Deep Learning!
 - Pytorch : <http://pytorch.org/docs/0.3.1/nn.html#recurrent-layers>
 - Tensorflow: <https://www.tensorflow.org/tutorials/recurrent>
 - Torch7: <https://github.com/jcjohnson/torch-rnn>
 - Blocks (Theano): <http://blocks.readthedocs.io/en/latest/rnn.html>
 - Keras : <https://keras.io/layers/recurrent/>
 - Etc ...

- Blog de Christopher Olah sur les LSTMs:
<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- Publication distill Christopher Olah sur les mécanismes d'attention:
<https://distill.pub/2016/augmented-rnns/>
- Blog de Andrej Karpathy sur les RNNs:
<http://karpathy.github.io/2015/05/21/rnn-effectiveness/>
- The Deep Learning Book (Goodfellow et al.):
<http://www.deeplearningbook.org/>



1. Retour sur la première partie

2. RNNs génératifs

3. Mécanismes d'attention

4. Librairies et références

QUESTIONS ?