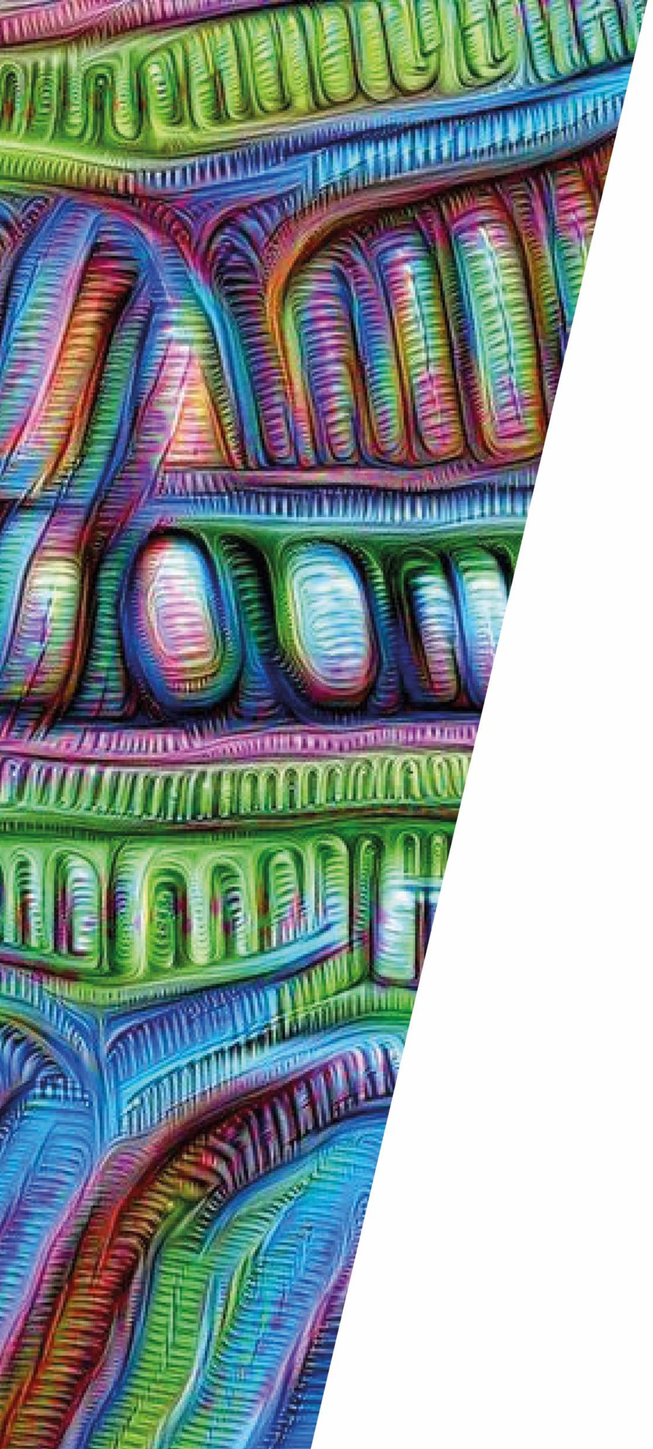


Bienvenue!

ÉCOLE D'ÉTÉ FRANCOPHONE EN APPRENTISSAGE PROFOND

21-25 août 2017

IVADO

HEC Montréal
Polytechnique Montréal
Université de Montréal

MILA

Institut
des algorithmes
d’apprentissage
de Montréal

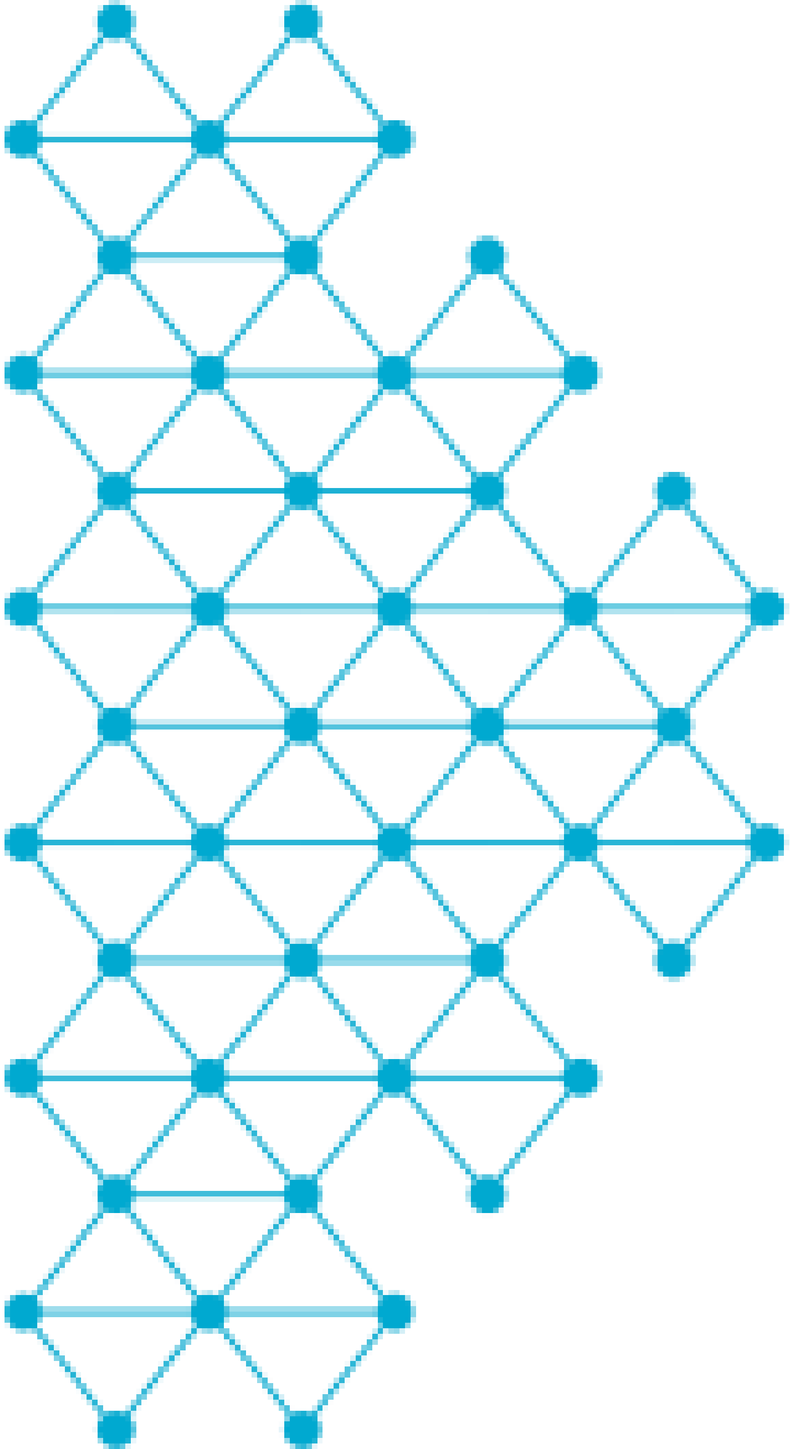
Réseaux Récurrents I

Ecole d’été IVADO

César Laurent

Plan

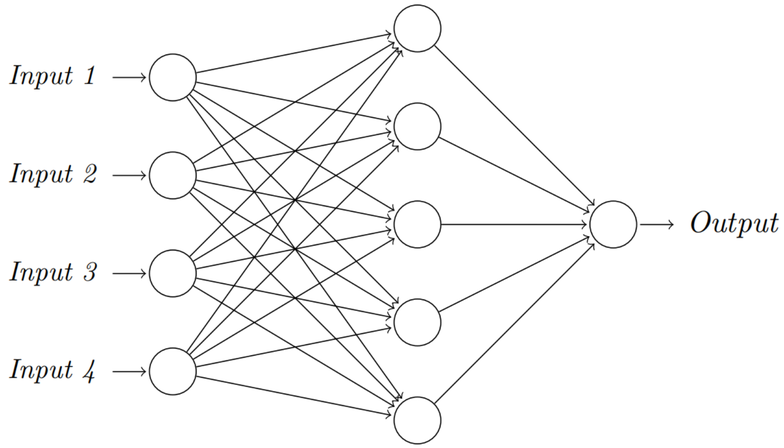
1. **Motivation**
2. **Introduction aux Réseaux de Neurones Récurrents (RNNs)**
3. **Entraînement des RNNs**
4. **Difficultés d’apprentissage**
5. **Architectures de RNNs**
6. **RNNs Génératifs**

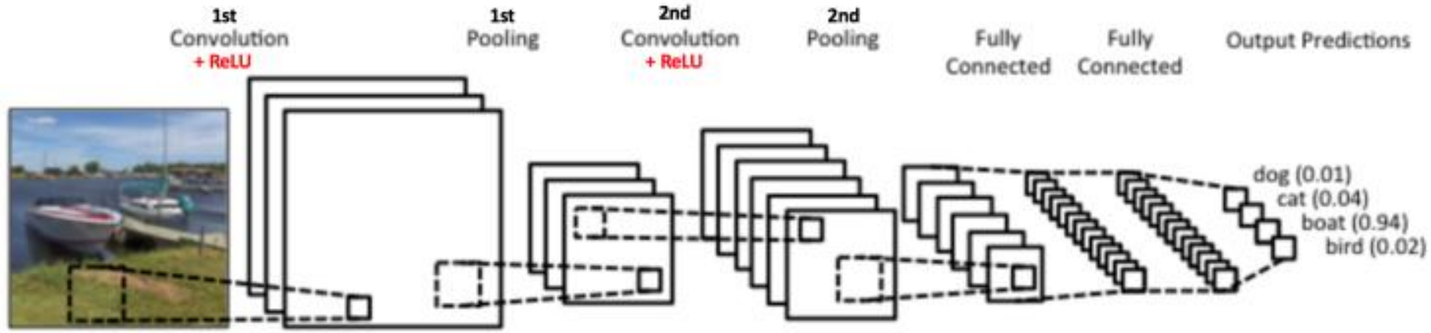


1. Motivation

Motivation

- Vous avez vu comment traîter des données de taille fixe et des images.





Comment traîter des **données séquentielles**?

Image: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets

Motivation

Reconnaissance de la parole

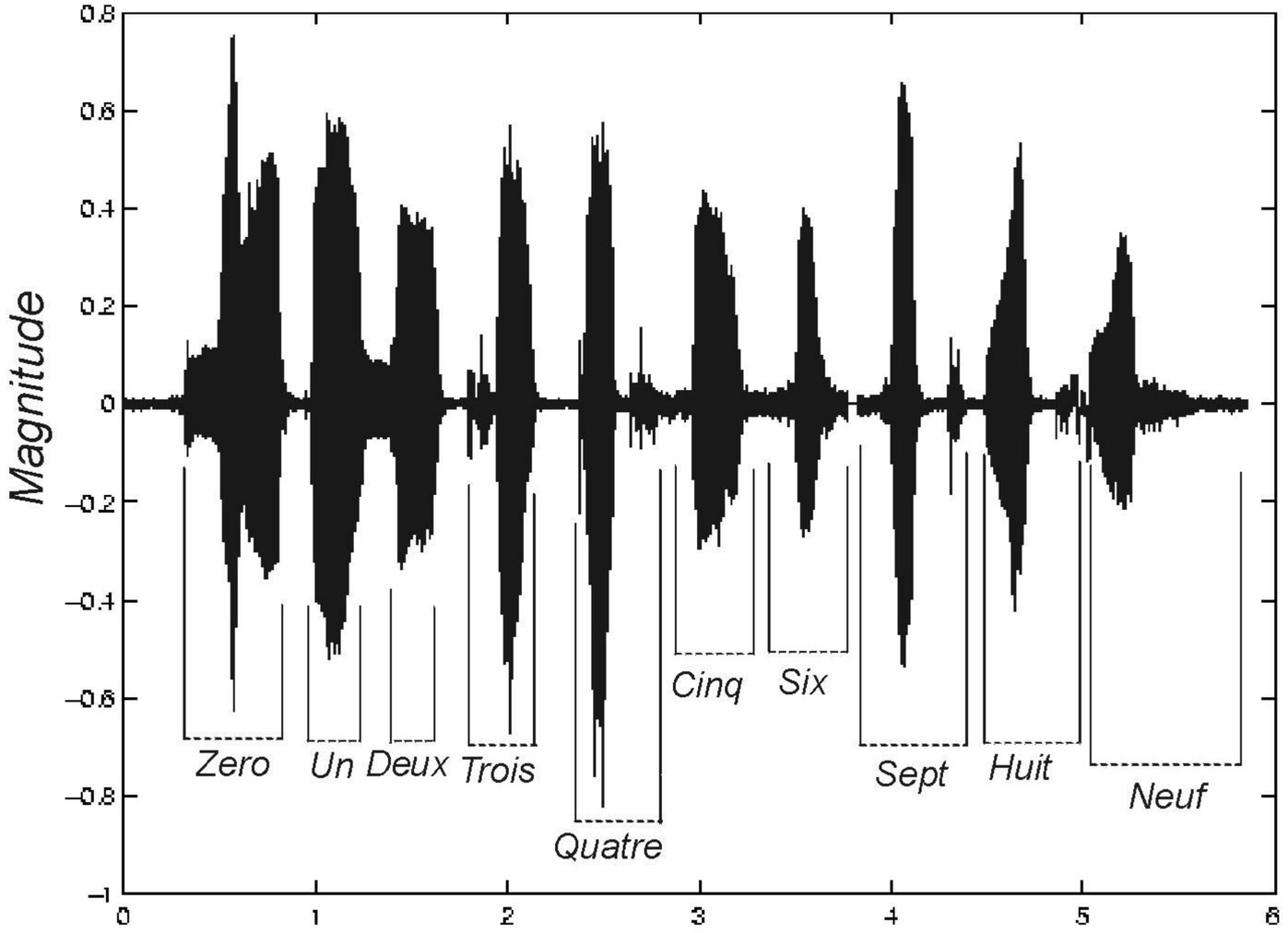
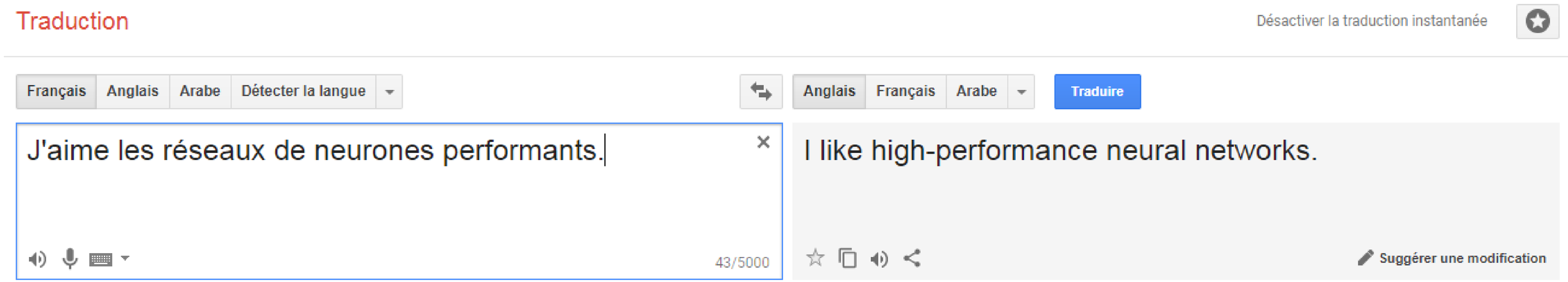


Image: http://poseidon.csd.auth.gr/LAB_RESEARCH/Latest/SecurityBiometrics.htm

Motivation

Traduction automatique



Motivation

Génération de légendes

A woman is throwing a frisbee in a park.

A stop sign is on a road with a mountain in the background.

A group of people sitting on a boat in the water.

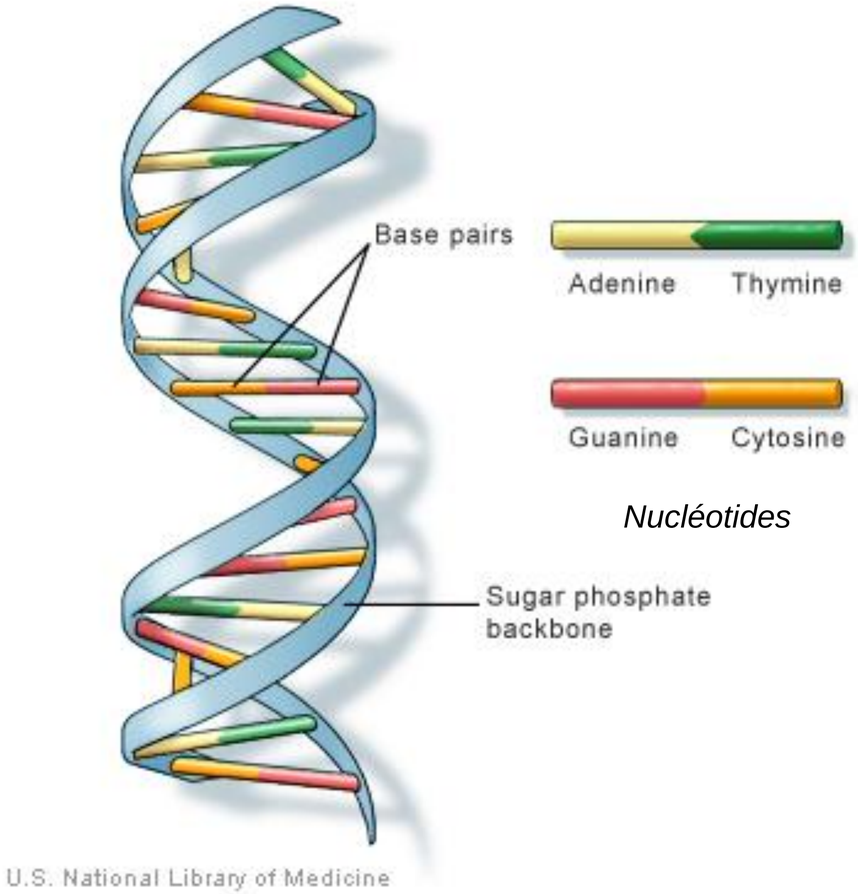
A giraffe standing in a forest with trees in the background.

Images: Xu et al., *Show, Attend and Tell: Neural Image Caption Generation with Visual Attention*

Motivation

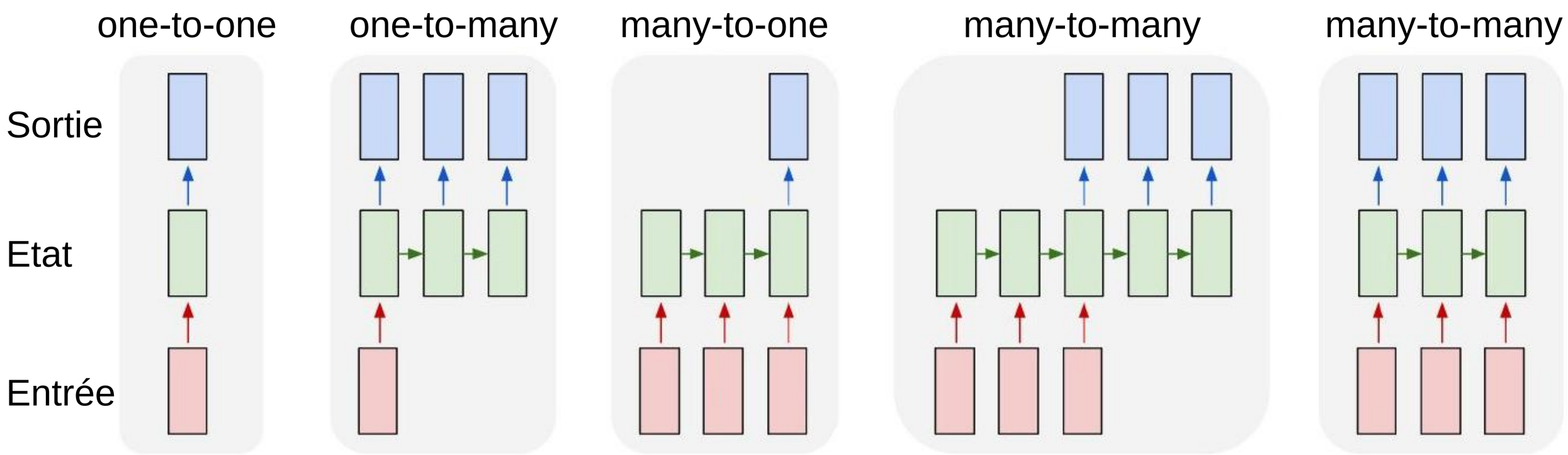
Plus d'exemples

- Vidéos
- Textes
- Séries financières
- Données biologiques (ADN, …)
- Signaux médicaux (IRMf, …)
- Etc…



Modélisation de séquences

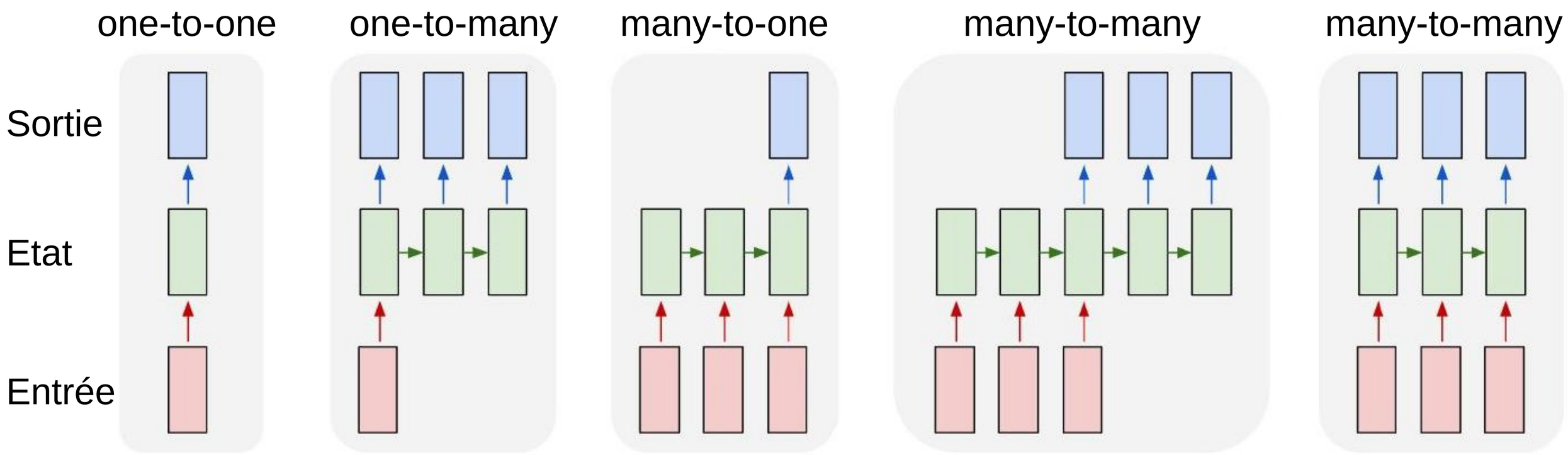
Différents types d'applications



Tiré de http://vision.stanford.edu/teaching/cs231n/slides/2016/winter1516_lecture10.pdf

Modélisation de séquences

Différents types d'applications



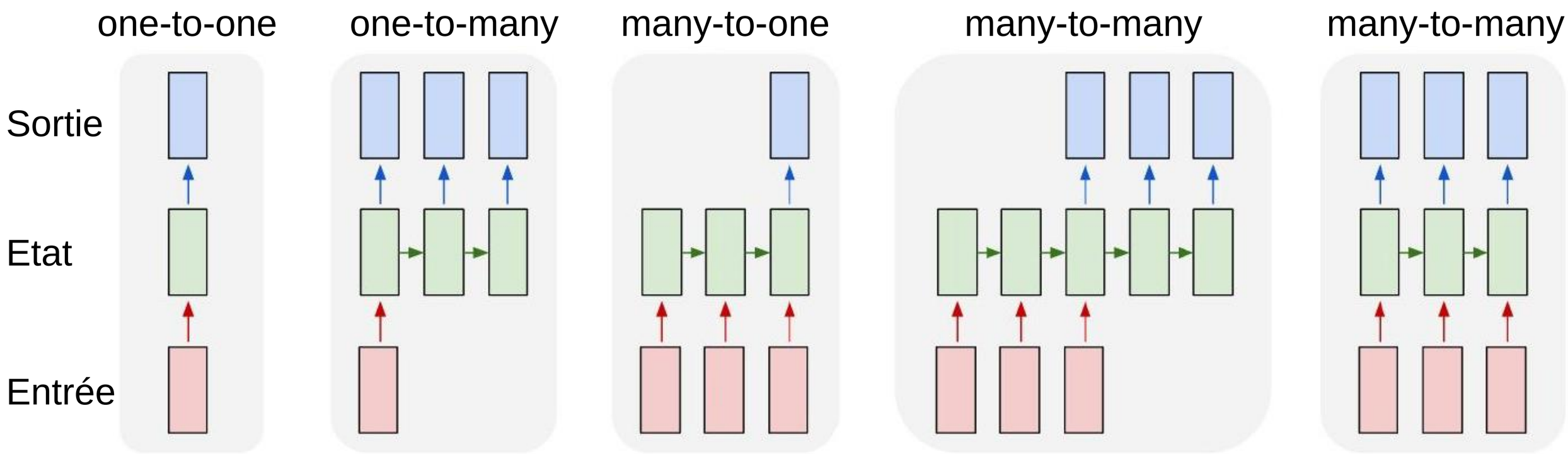
Classification d'objets

(Image → Classe)

Tiré de http://vision.stanford.edu/teaching/cs231n/slides/2016/winter1516_lecture10.pdf

Modélisation de séquences

Différents types d'applications



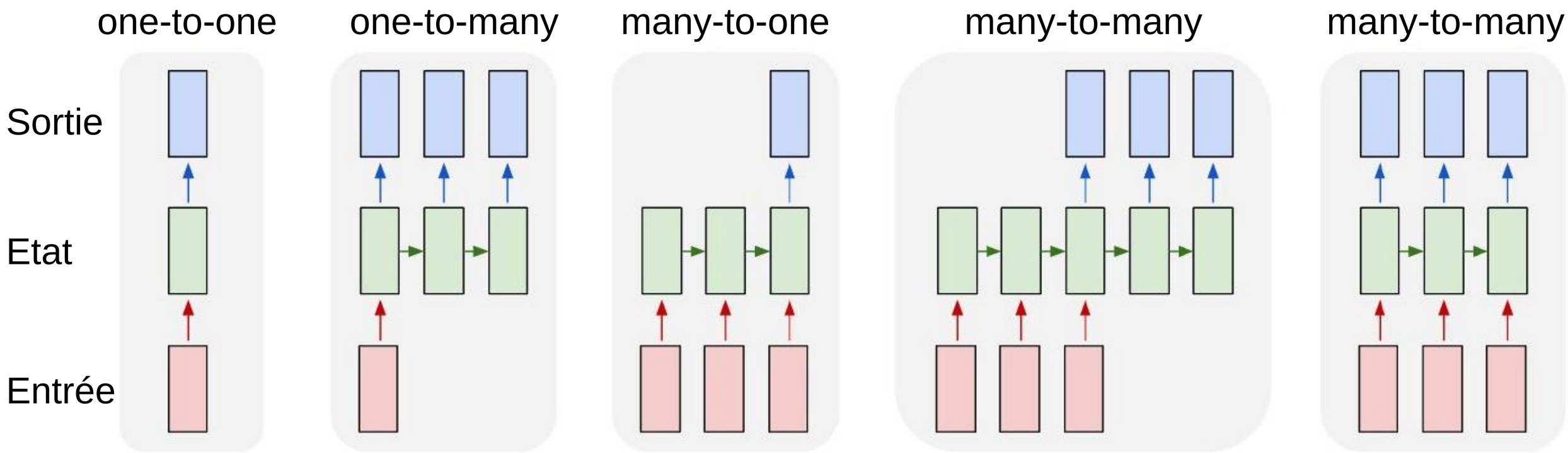
Génération de légende

(Image → Phrase)

Tiré de http://vision.stanford.edu/teaching/cs231n/slides/2016/winter1516_lecture10.pdf

Modélisation de séquences

Différents types d'applications



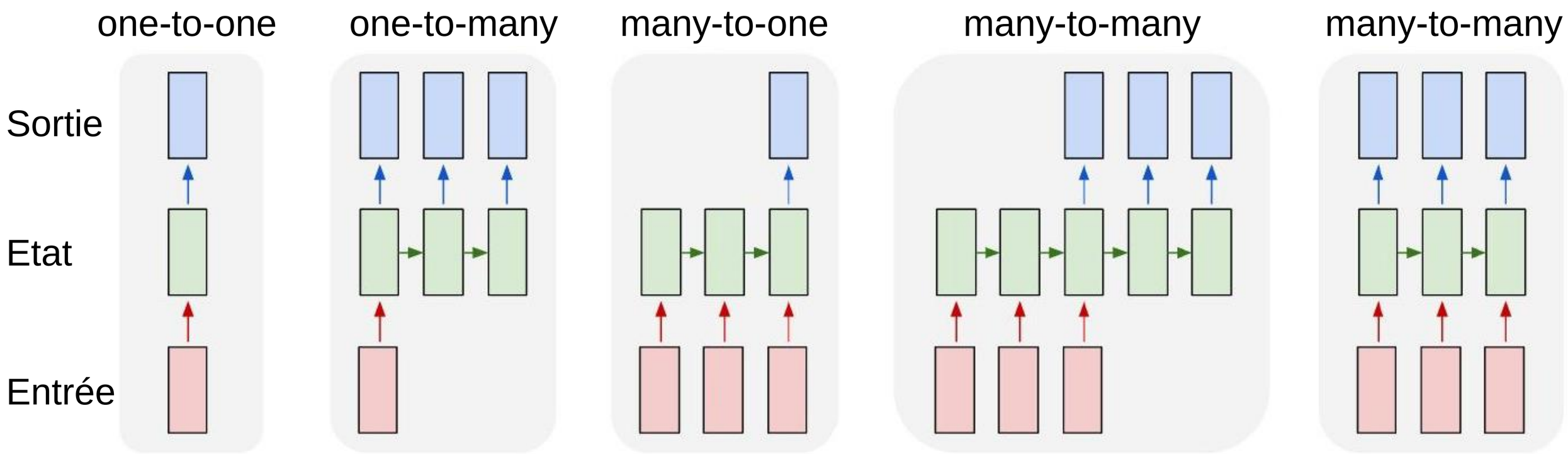
Analyse de sentiment

(Phrase → Classe)

Tiré de http://vision.stanford.edu/teaching/cs231n/slides/2016/winter1516_lecture10.pdf

Modélisation de séquences

Différents types d'applications



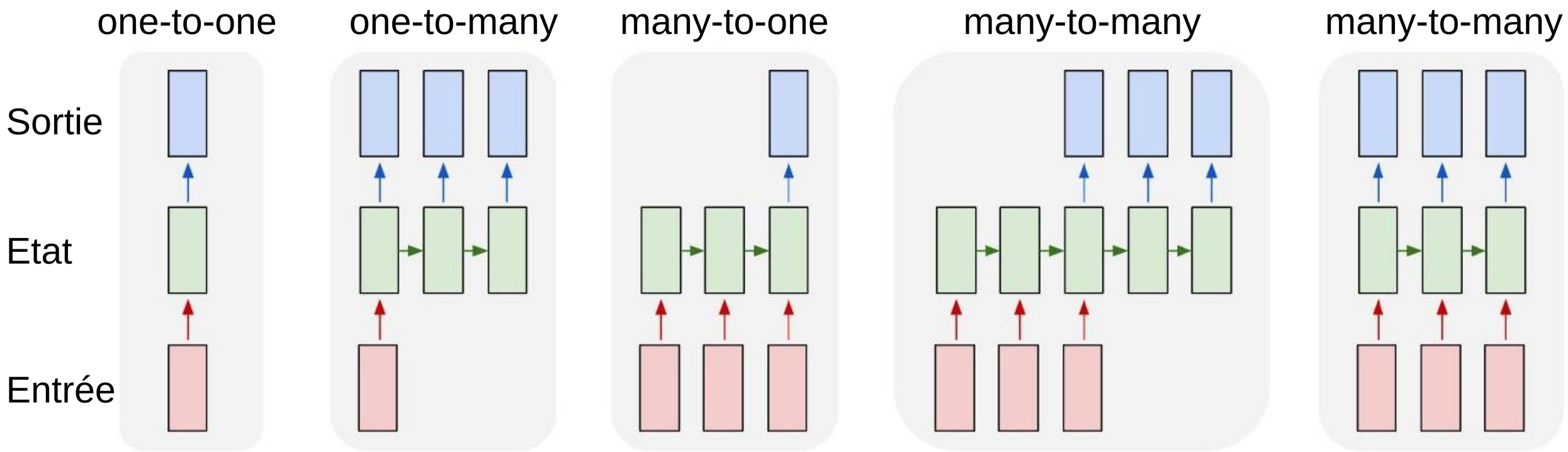
Traduction automatique

(Phrase → Phrase)

Tiré de http://vision.stanford.edu/teaching/cs231n/slides/2016/winter1516_lecture10.pdf

Modélisation de séquences

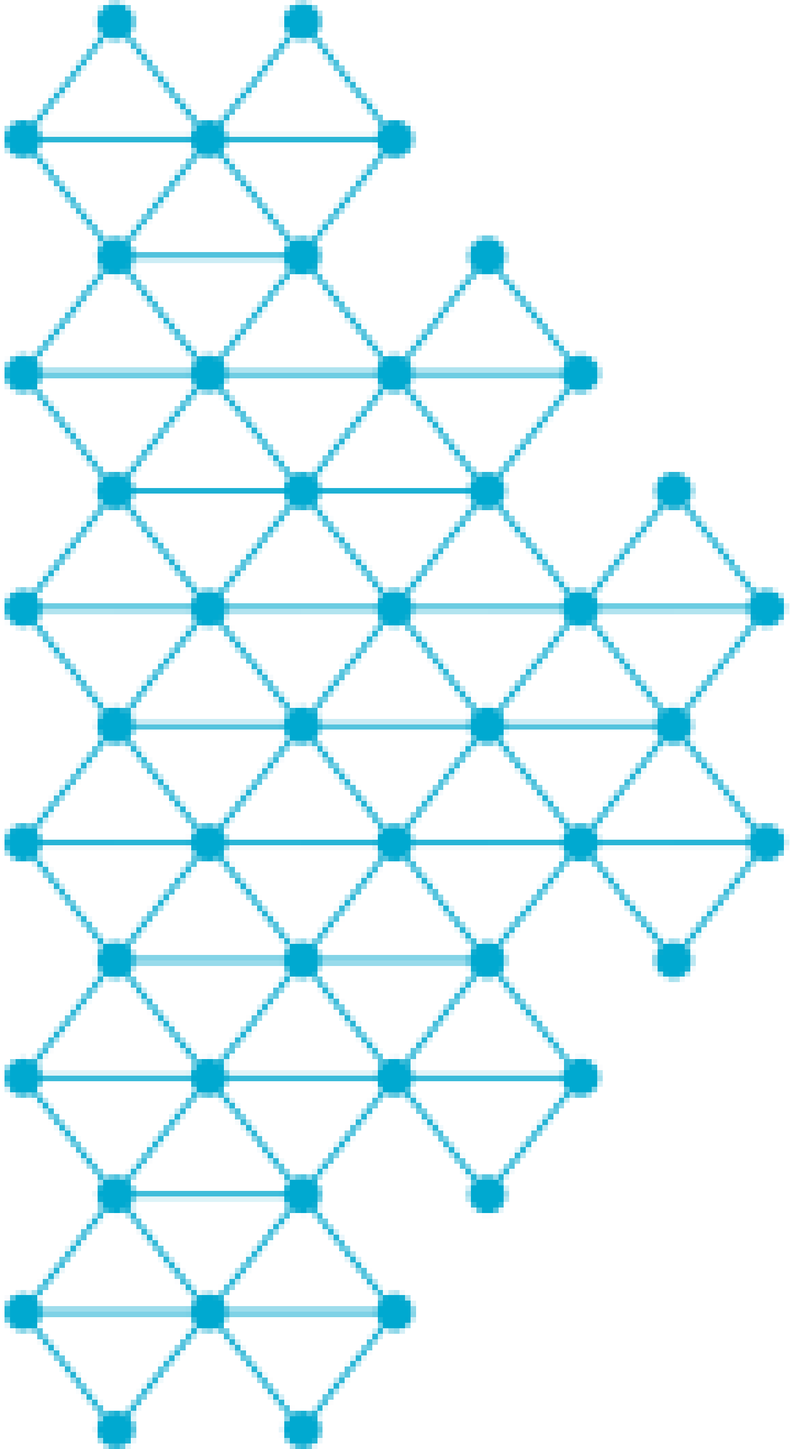
Différents types d'applications



Reconnaissance de la parole

(Son → Phrase)

Tiré de http://vision.stanford.edu/teaching/cs231n/slides/2016/winter1516_lecture10.pdf



2. Introduction aux RNNs

Réseaux Récurrents

Introduction

Un RNN applique une fonction à une **séquence d'entrée** $[x_1, x_2, \ldots, x_T]$, pour produire une **séquence de sortie** $[y_1, y_2, \ldots, y_T]$, en maintenant un **état interne** $[h_1, h_2, \ldots, h_T]$.

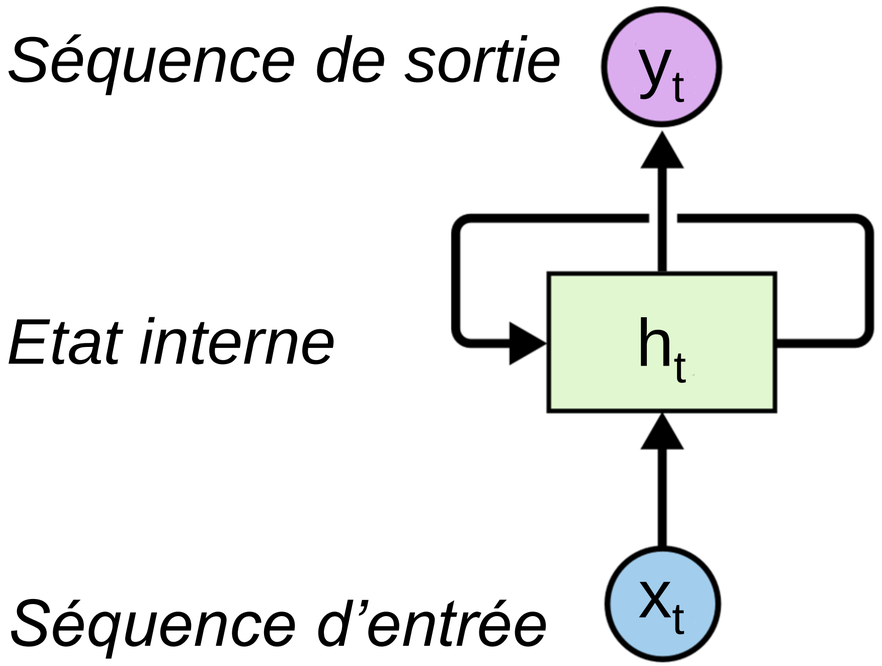


Image from Christopher Olah's blog

Réseaux Récurrents

Un exemple simple

- La version la plus simple:

$$h_t = \tanh(Ux_t + Wh_{t-1})$$

$$y_t = f(Vh_t)$$

- W, U et V sont les paramètres du réseau.
  - Ils sont **partagés** à travers le temps.
- h_{-1} peut également être un paramètre.

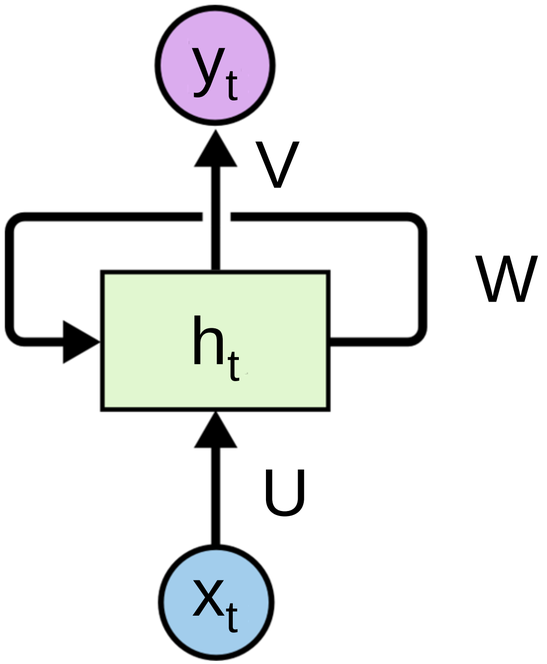
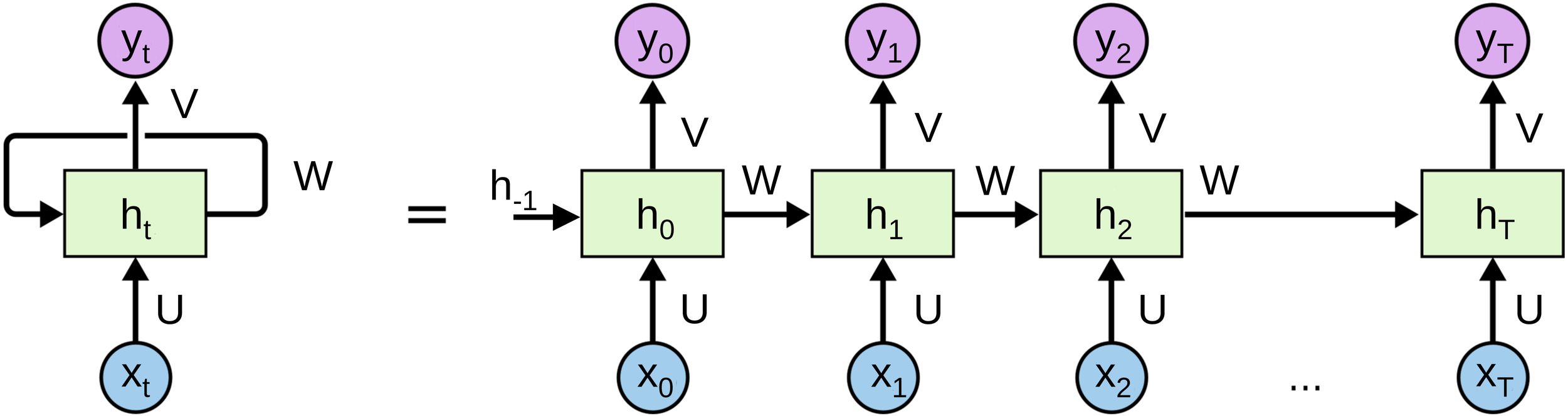


Image from Christopher Olah's blog

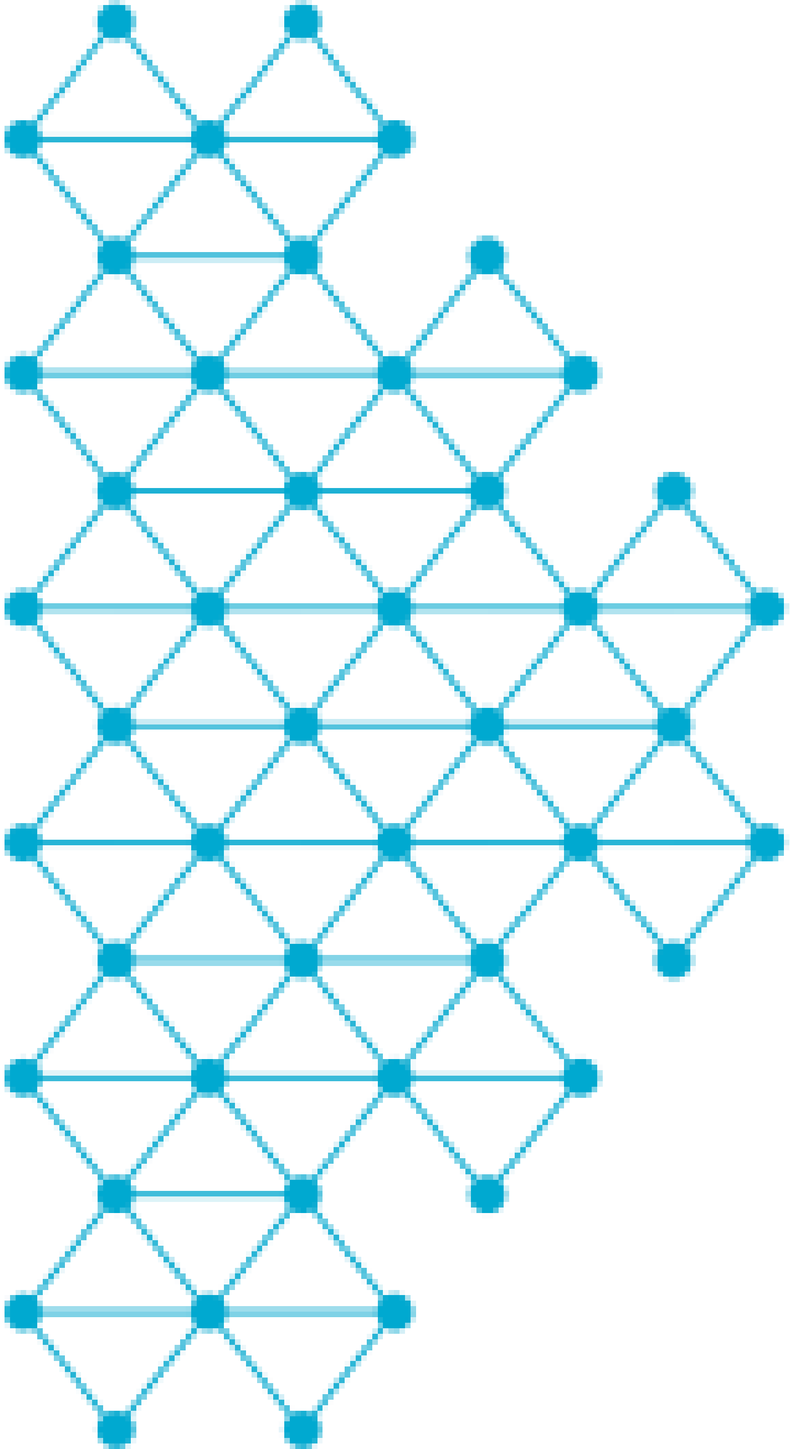
Réseaux Récurrents

Exemple déroulé dans le temps



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

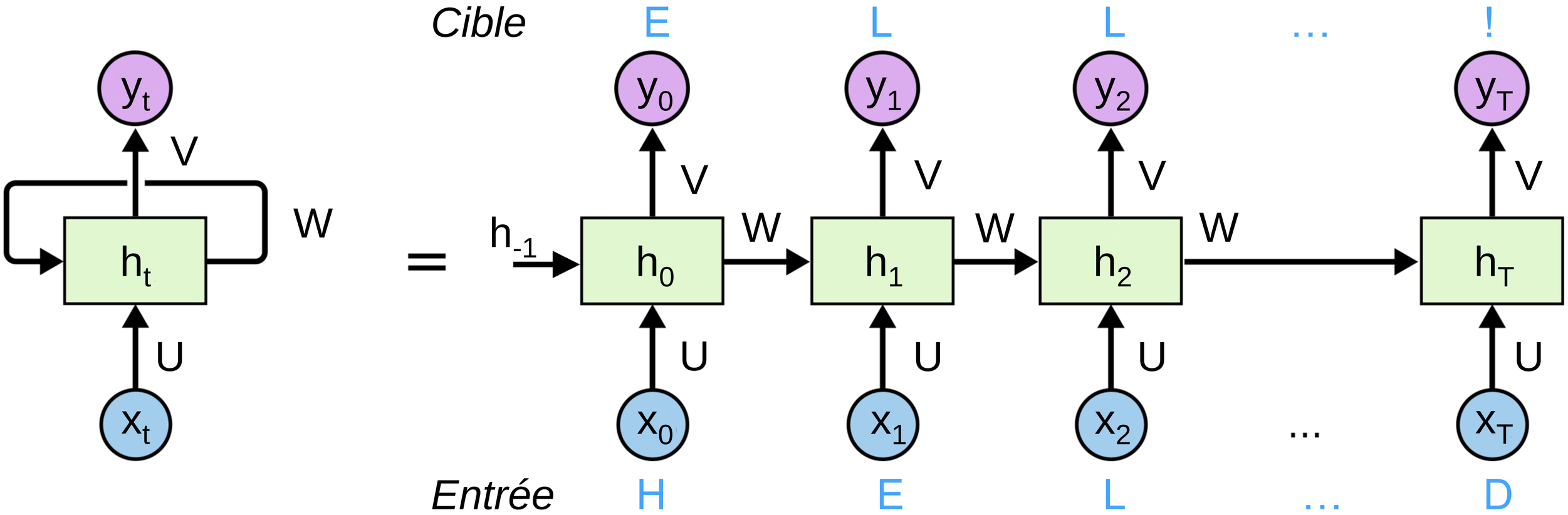
Image from Christopher Olah's blog



3. Entraînement des RNNs

Réseaux Récurrents

Exemple de tâche: Prédiction du prochain caractère



Erreur: Entropie croisée à chaque temps.

Image from Christopher Olah's blog

Rétropropagation à travers le temps

Introduction

- Calcul de l'erreur globale:

$$E = \sum_{t=0}^{T} E_t$$

- Rétropropagation classique pour adapter les paramètres.

- Prenons l'exemple de U:

$$\frac{\partial E}{\partial U} = \sum_{t=0}^{T} \frac{\partial E_t}{\partial U}$$

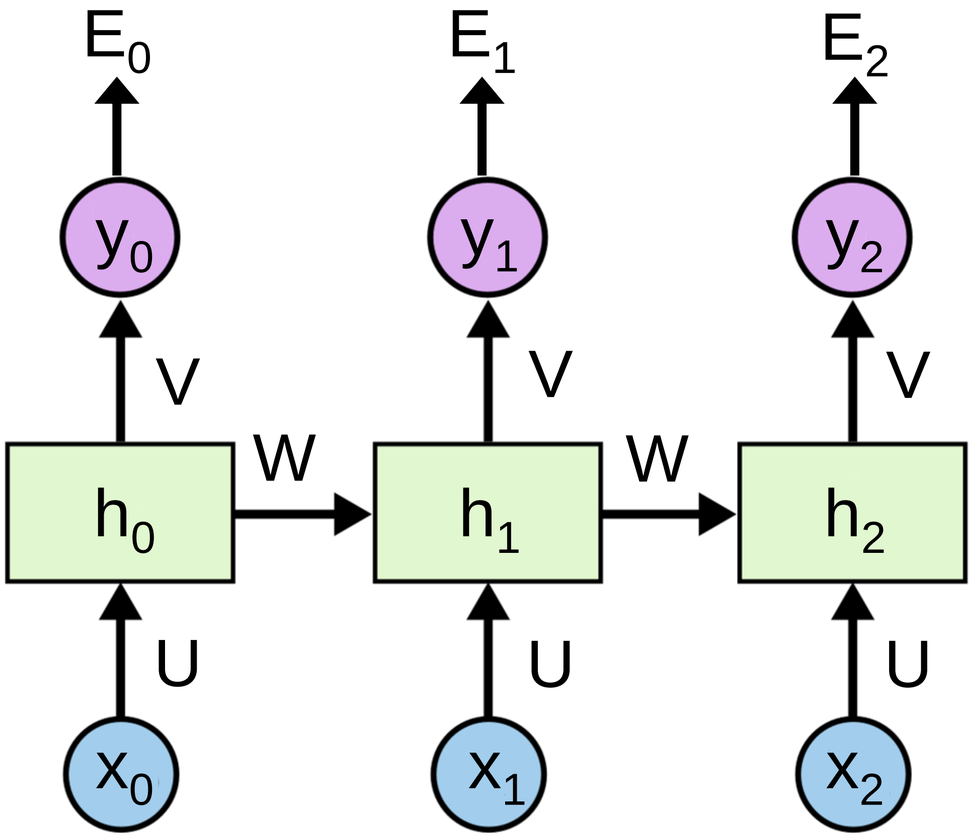
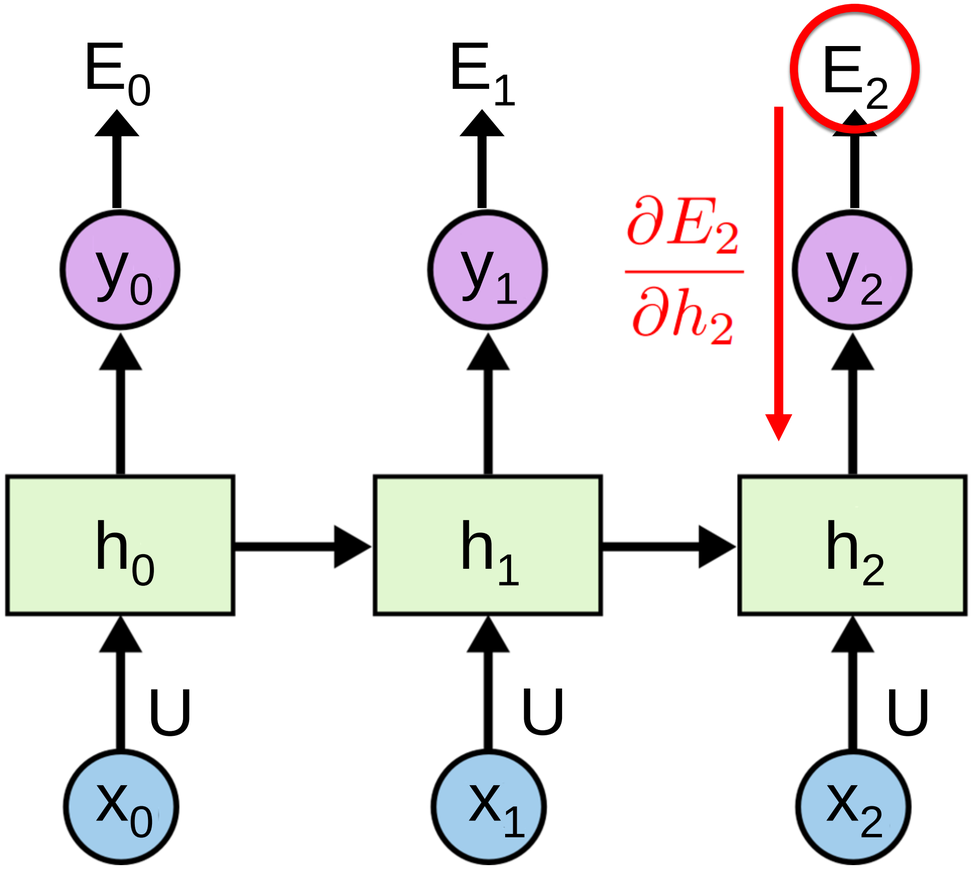


Image from Christopher Olah's blog

Rétropropagation à travers le temps

Exemple: Calcul du gradient sur U

- Pour calculer dE/dU, calculons d'abord dE_2/dU.
  - La première étape est de propager l'erreur jusqu'à l'état interne h_2.



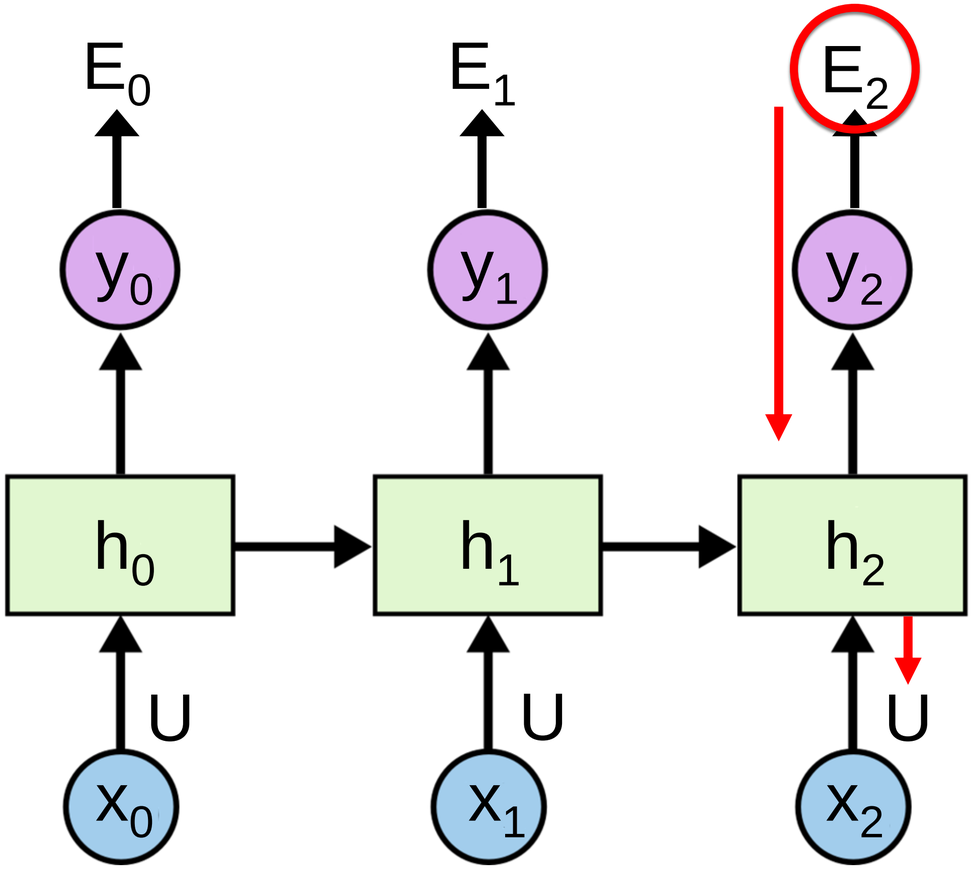
$$\frac{\partial E_2}{\partial U} = \frac{\partial E_2}{\partial h_2} \cdots$$

Image from Christopher Olah's blog

Rétropropagation à travers le temps

Exemple: Calcul du gradient sur U

- Pour calculer dE/dU, calculons d'abord dE_2/dU.
  - L'erreur peut ensuite être propagée sur U au 2ème pas de temps.



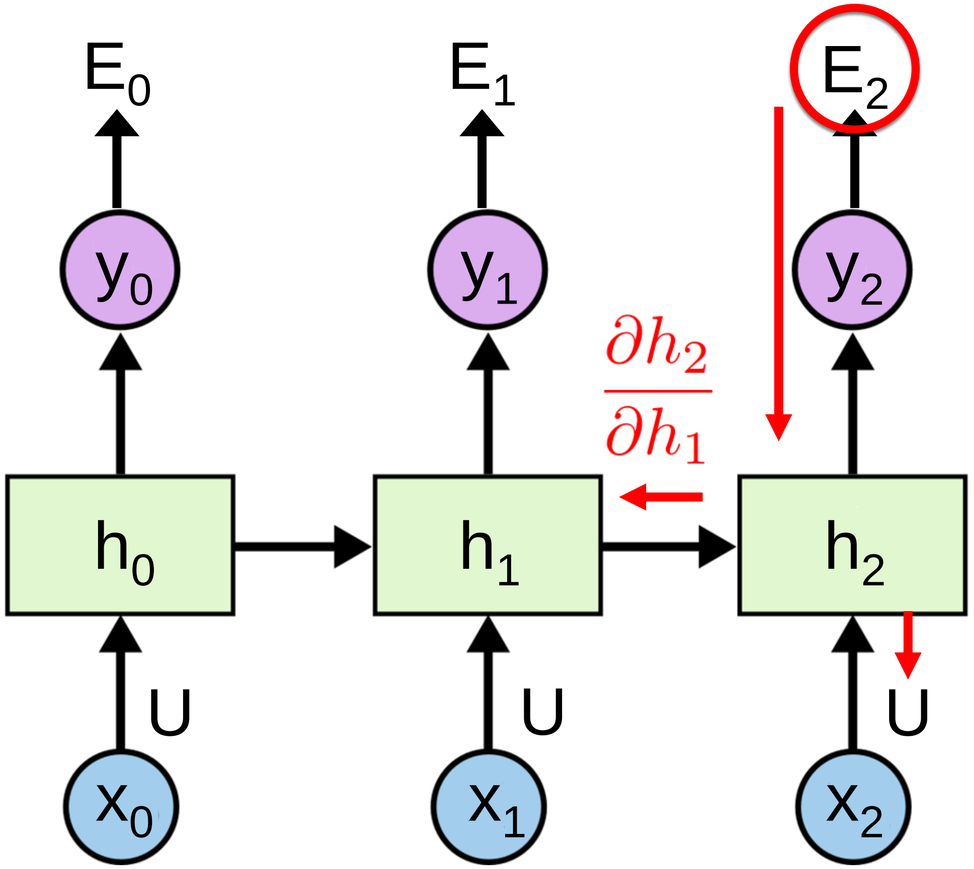
$$\frac{\partial E_2}{\partial U} = \frac{\partial E_2}{\partial h_2} \left(x_2^T \cdots \right.$$

Image from Christopher Olah's blog

Rétropropagation à travers le temps

Exemple: Calcul du gradient sur U

- Pour calculer dE/dU, calculons d'abord dE_2/dU.
  - Mais U intervient également dans le calcul de h_1 et h_0.
  - Il faut donc rétropropager l'erreur à travers le temps.



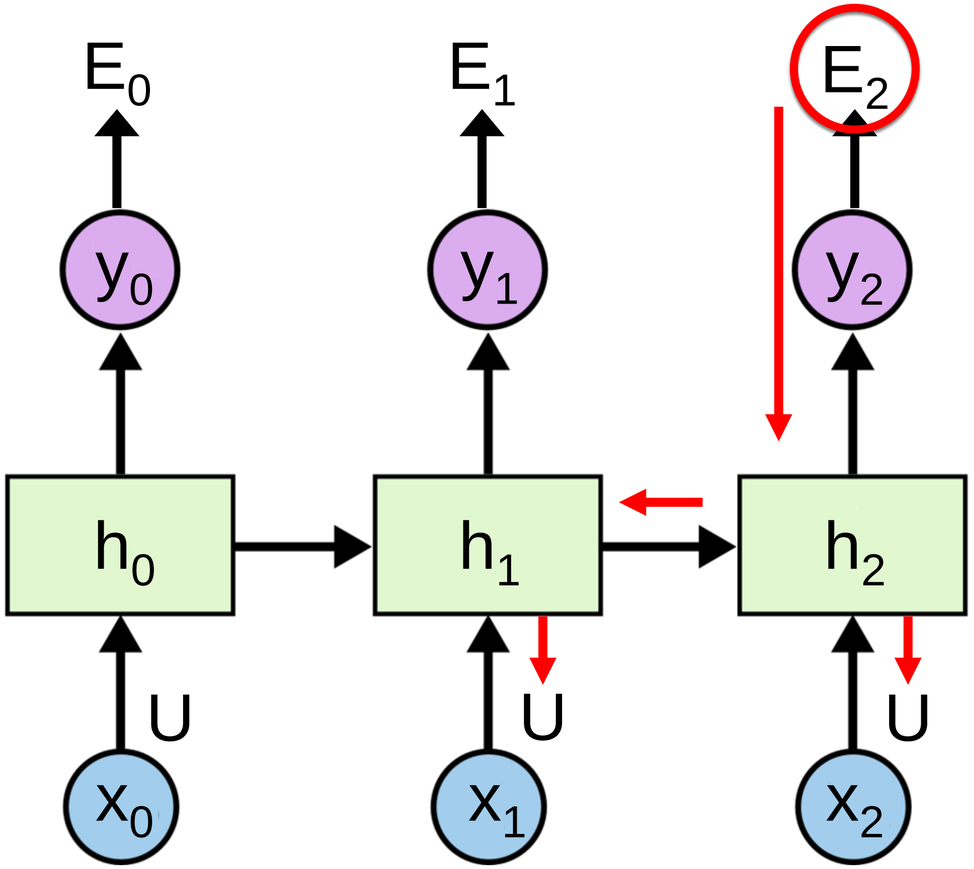
$$\frac{\partial E_2}{\partial U} = \frac{\partial E_2}{\partial h_2}\left(x_2^T + \frac{\partial h_2}{\partial h_1}(\cdots\right.$$

Image from Christopher Olah's blog

Rétropropagation à travers le temps

Exemple: Calcul du gradient sur U

- Pour calculer dE/dU, calculons d'abord dE_2/dU.
  - L'erreur peut ensuite être propagée sur U au 1ème pas de temps.



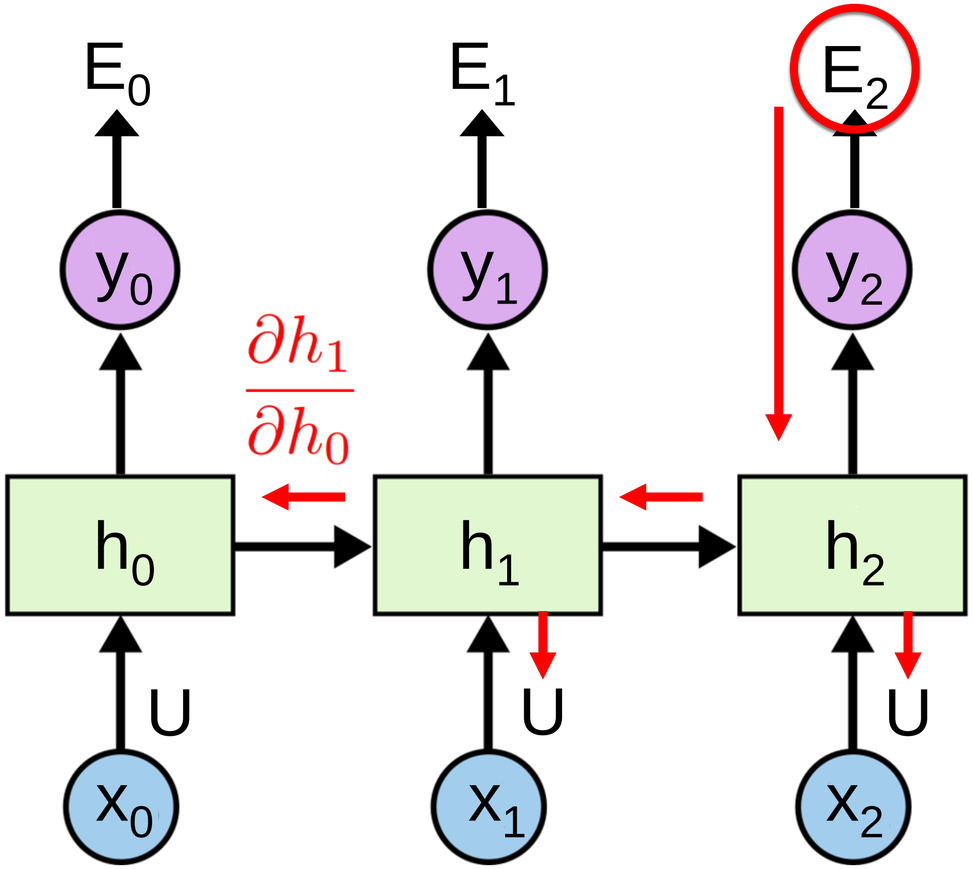
$$\frac{\partial E_2}{\partial U} = \frac{\partial E_2}{\partial h_2}\left(x_2^T + \frac{\partial h_2}{\partial h_1}(x_1^T \cdots\right.$$

Image from Christopher Olah's blog

Rétropropagation à travers le temps

Exemple: Calcul du gradient sur U

- Pour calculer dE/dU, calculons d'abord dE_2/dU.
  - Mais U intervient également dans le calcul h_0.
  - Il faut donc encore rétropropager l'erreur à travers le temps.



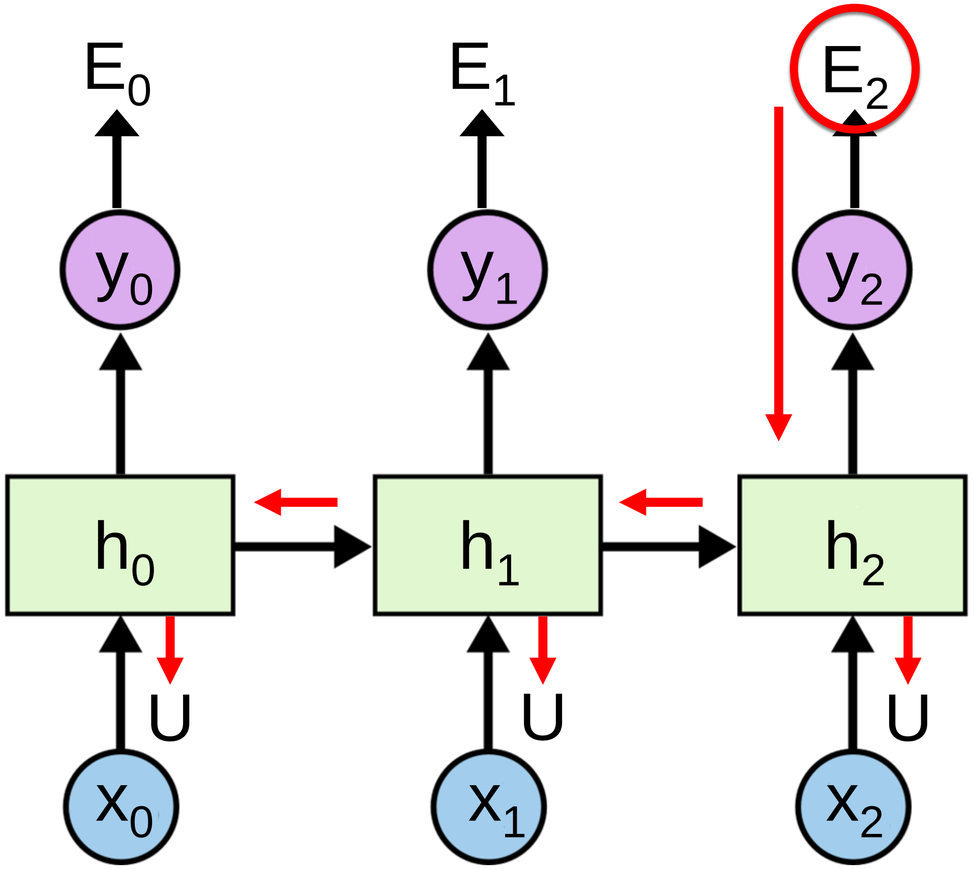
$$\frac{\partial E_2}{\partial U} = \frac{\partial E_2}{\partial h_2}\left(x_2^T + \frac{\partial h_2}{\partial h_1}\left(x_1^T + \frac{\partial h_1}{\partial h_0}\cdots\right.\right.$$

Image from Christopher Olah's blog

Rétropropagation à travers le temps

Exemple: Calcul du gradient sur U

- Pour calculer dE/dU, calculons d'abord dE_2/dU.
  - L'erreur peut ensuite être propagée sur U au 1ème pas de temps.
  - La rétropropagation dE_2/dU est maintenant complète!



$$\frac{\partial E_2}{\partial U} = \frac{\partial E_2}{\partial h_2}\left(x_2^T + \frac{\partial h_2}{\partial h_1}\left(x_1^T + \frac{\partial h_1}{\partial h_0}x_0^T\right)\right)$$

Image from Christopher Olah's blog

Rétropropagation à travers le temps

Exemple: Calcul du gradient sur U

- Les mêmes opérations sont effectuées pour calculer dE_1/dU

1. Rétropropager l'erreur sur h_1
2. Calculer dh_1/dU au pas de temps 1
3. Rétropropager l'erreur dans le temps sur h_0
4. Calculer dh_0/dU

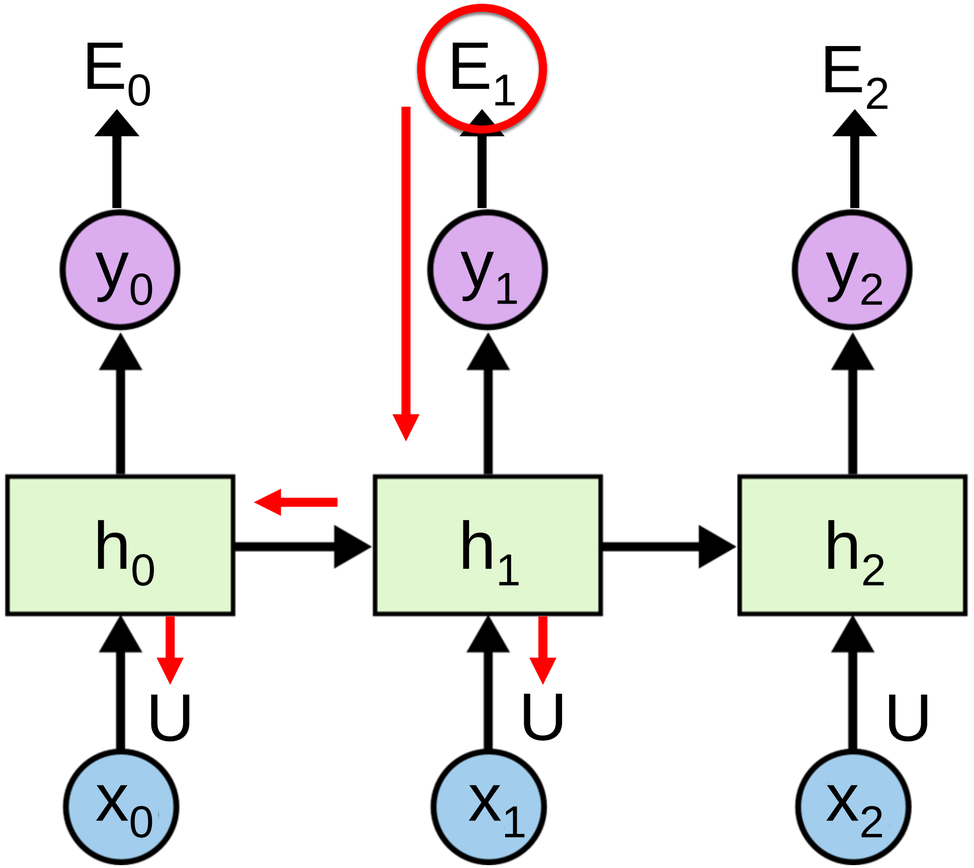


Image from Christopher Olah's blog

Rétropropagation à travers le temps

Exemple: Calcul du gradient sur U

- Les mêmes opérations sont effectuées pour calculer dE_0/dU

1. Rétropropager l'erreur sur h_0
2. Calculer dh_0/dU

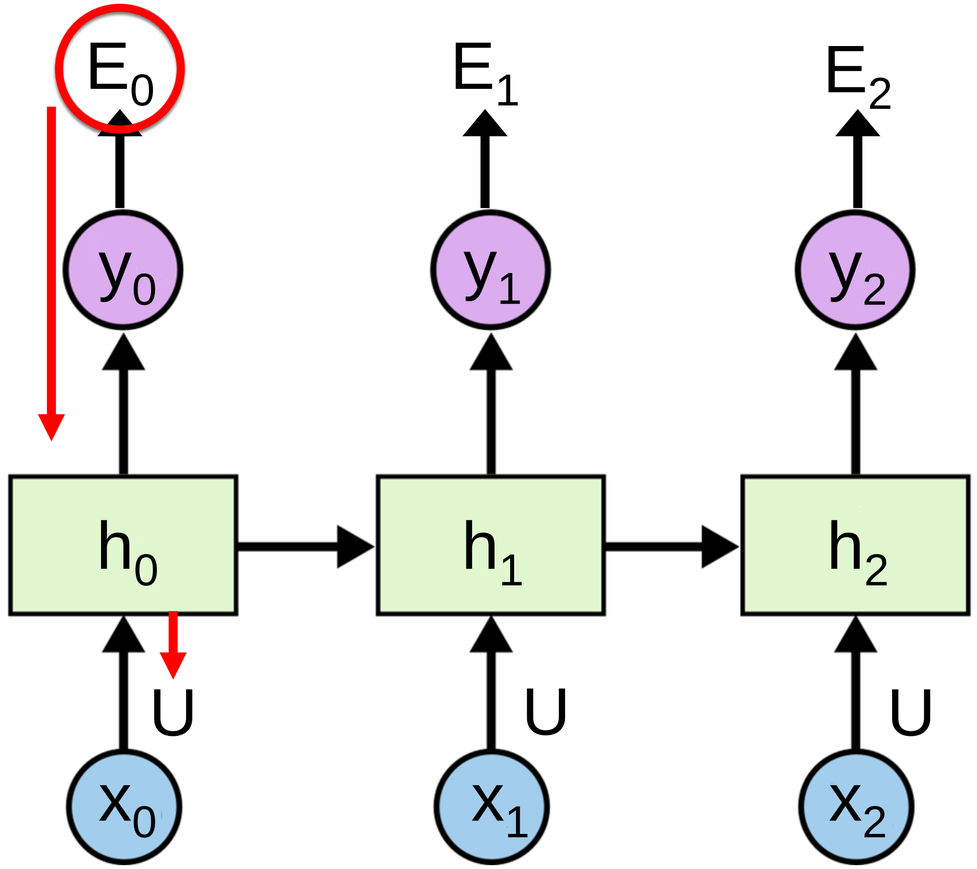


Image from Christopher Olah's blog

Rétropropagation à travers le temps

Exemple: Calcul du gradient sur U

- Toutes les contributions sont sommées pour obtenir le gradient sur U:

$$\frac{\partial E}{\partial U} = \sum_{t=0}^{T} \frac{\partial E_t}{\partial U}$$

- Les gradients sur les autres paramètres sont calculés de la même manière.

$$\frac{\partial E}{\partial V} = \sum_{t=0}^{T} \frac{\partial E_t}{\partial V} \qquad \frac{\partial E}{\partial W} = \sum_{t=0}^{T} \frac{\partial E_t}{\partial W}$$

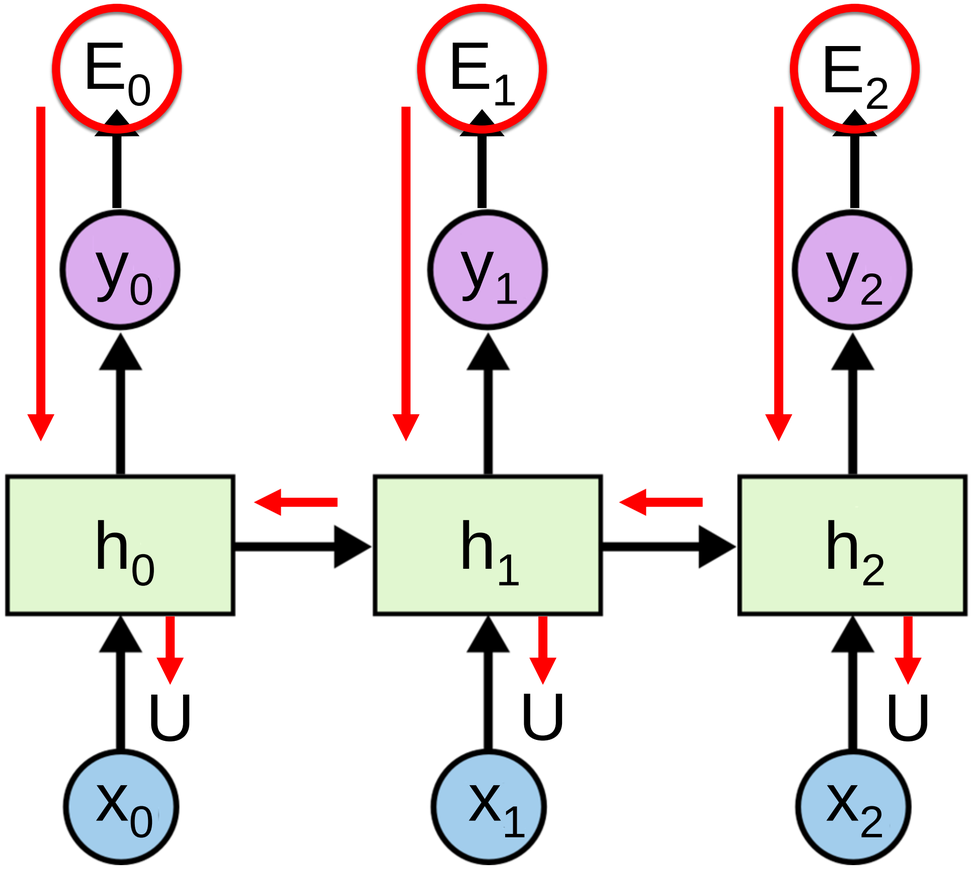
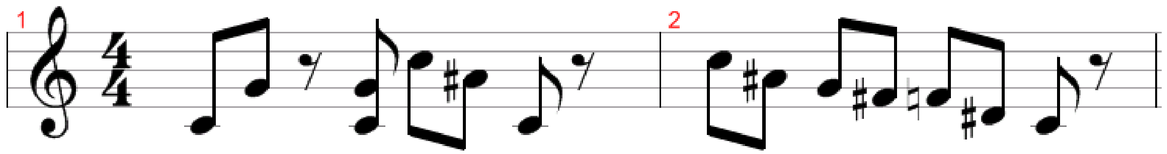


Image from Christopher Olah's blog

Exemple: Genre musical

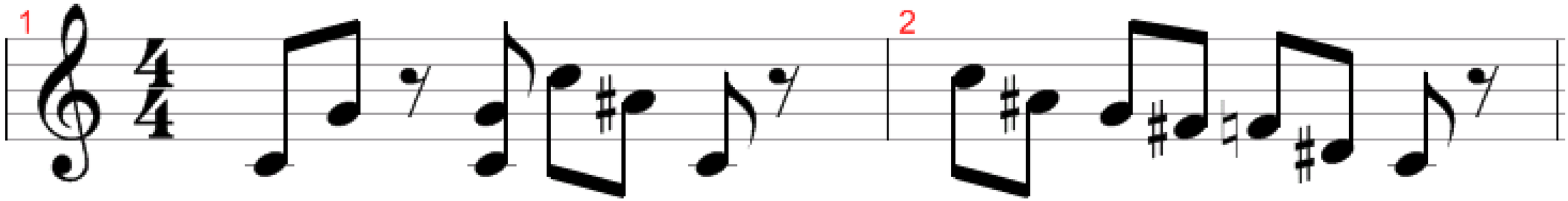
Présentation de la tâche

But: Reconnaitre le genre musical à partir de la partition

- Données: 500 partitions de musique (durée variable):
  - x: 
- 3 Classes: Blues, Rock, Classique
  - c: (1, 0, 0) ou (0, 1, 0) ou (0, 0, 1)

Exemple: Genre musical

Représentation des données

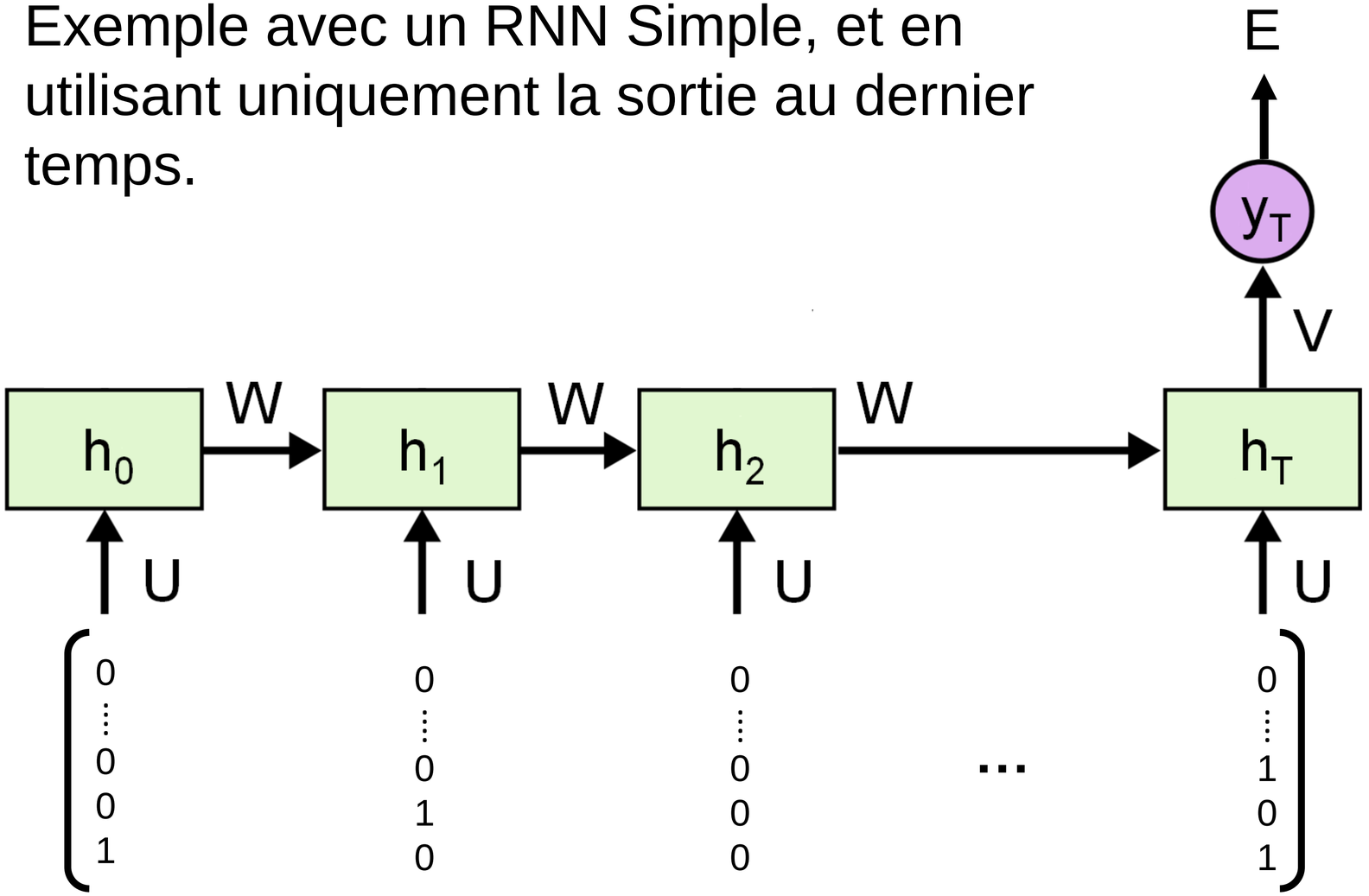


0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	1	0	1	0	1	0	0	1	1	0	1	0

Exemple: Genre musical

Réseau récurrent

Exemple avec un RNN Simple, et en utilisant uniquement la sortie au dernier temps.



$$E = \text{cross\_entropy}(y_T, c)$$

$$y_T = \text{Softmax}(V h_T + d)$$

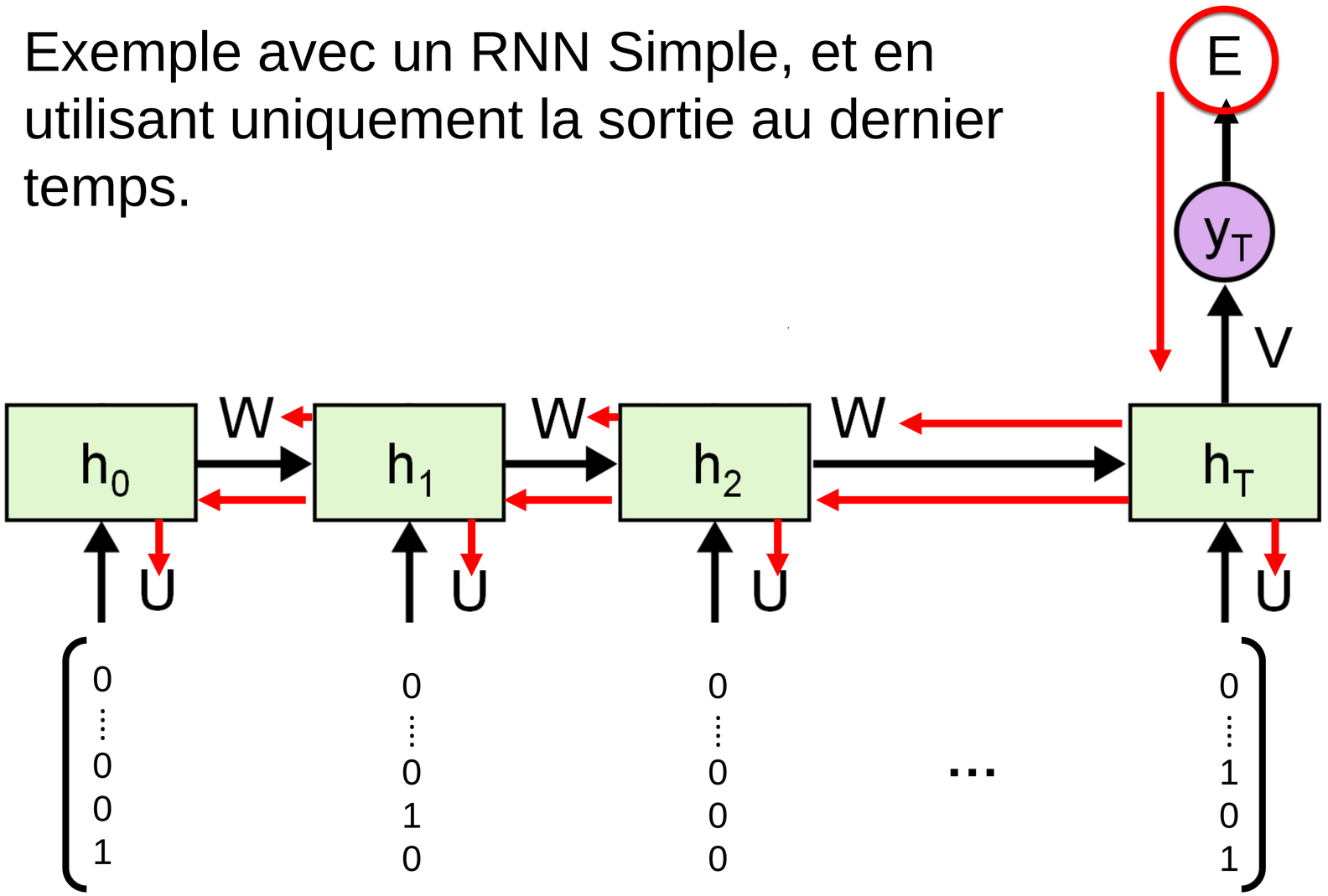
$$h_t = \tanh(U x_t + W h_{t-1} + b)$$

$$x$$

Exemple: Genre musical

Rétropropagation des gradients

Exemple avec un RNN Simple, et en utilisant uniquement la sortie au dernier temps.



$$E = \text{cross\_entropy}(y_T, c)$$

$$y_T = \text{Softmax}(V h_T + d)$$

$$h_t = \tanh(U x_t + W h_{t-1} + b)$$

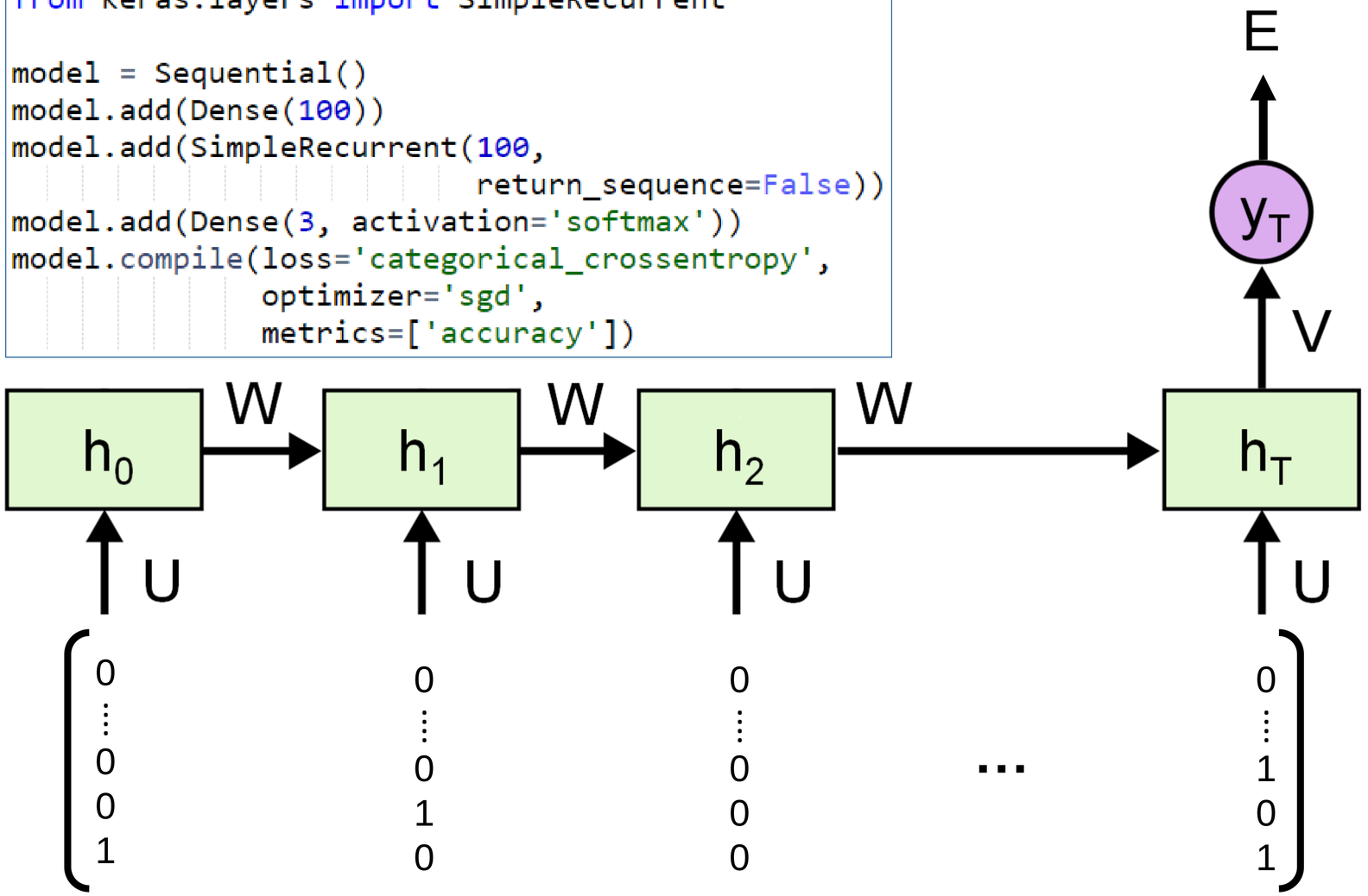
$$x$$

Exemple: Genre musical

Réseau récurrent et code

```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import SimpleRecurrent

model = Sequential()
model.add(Dense(100))
model.add(SimpleRecurrent(100,
                          return_sequence=False))
model.add(Dense(3, activation='softmax'))
model.compile(loss='categorical_crossentropy',
              optimizer='sgd',
              metrics=['accuracy'])
```



$$E = \text{cross\_entropy}(y_T, c)$$

$$y_T = \text{Softmax}(V h_T + d)$$

$$h_t = \tanh(U x_t + W h_{t-1} + b)$$

$$x$$

Réseaux Récurrents

Autres exemples

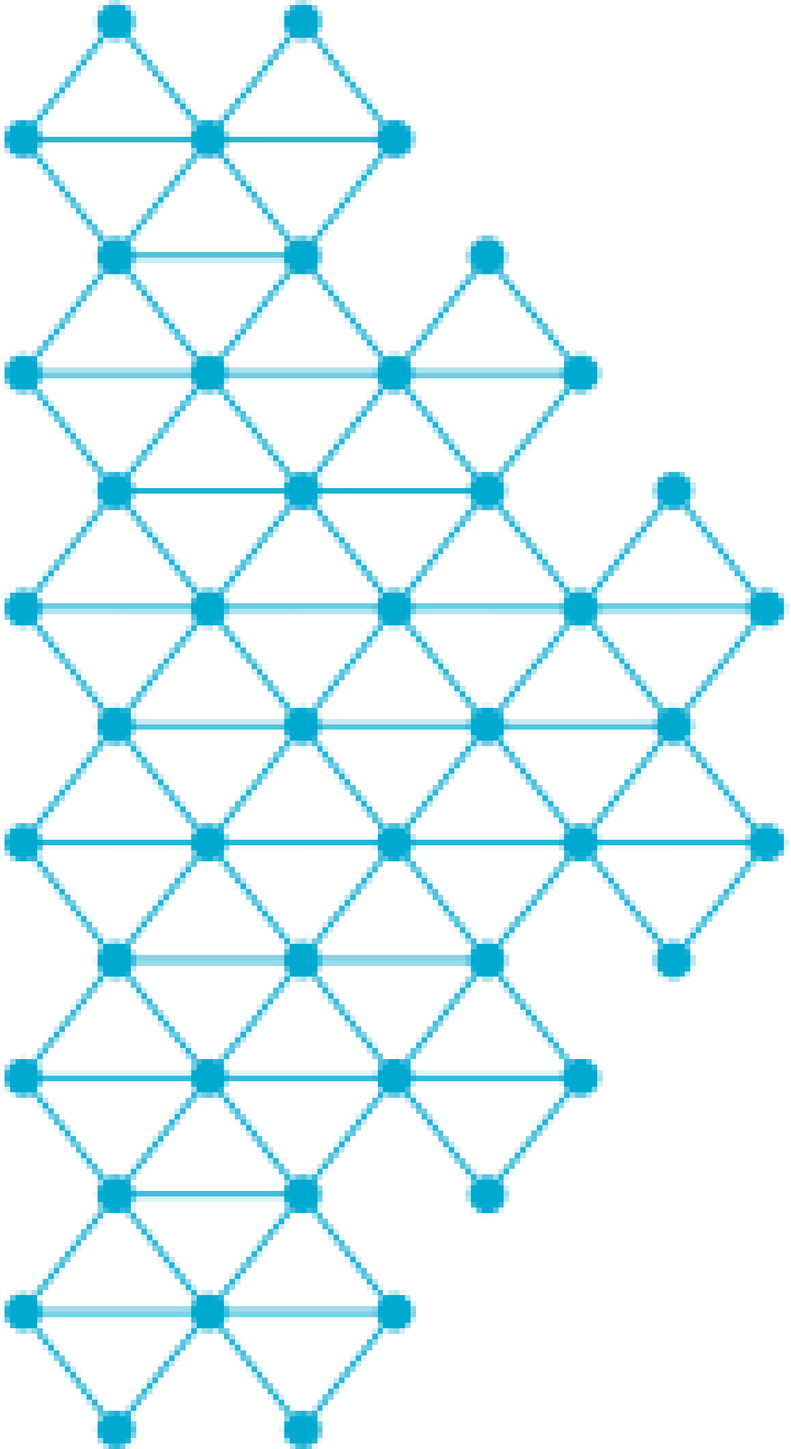
D’autres exemples d’application simple:

- Analyse de sentiment (texte)
- Détection d’arythmie cardiaque (ECG)
- Classification de séquences

Rétropropagation à travers le temps

Librairies d'apprentissage profond

- Si vous n'avez pas tout compris:
  - Les librairies d'apprentissage profond calculent les gradients automatiquement pour vous!
- **Attention:** Il y a quand même quelques difficultés d'apprentissage qui peuvent survenir.



4. Difficultés d’Apprentissage

Dépendances à long terme

Exemple

Qui est allé à Paris?

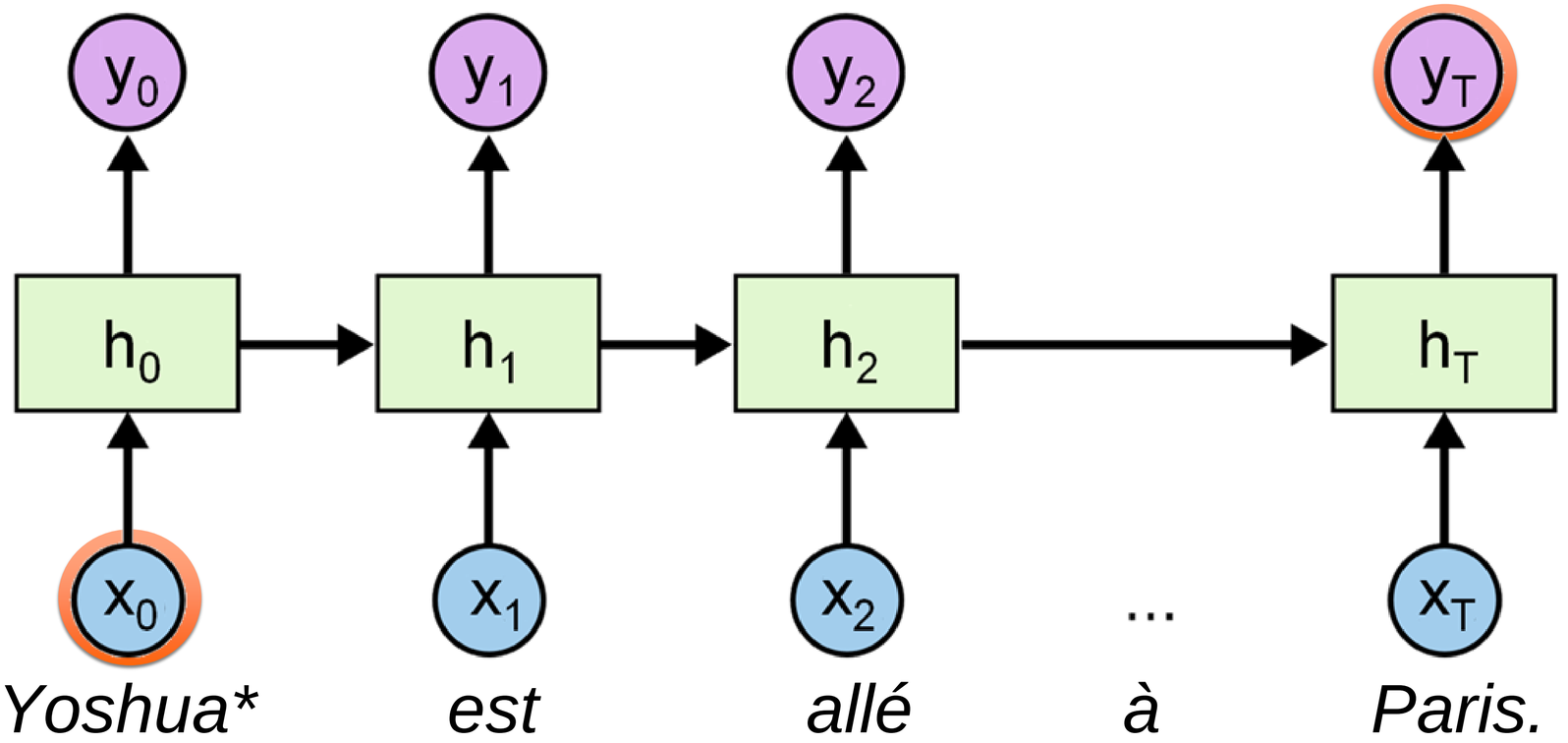


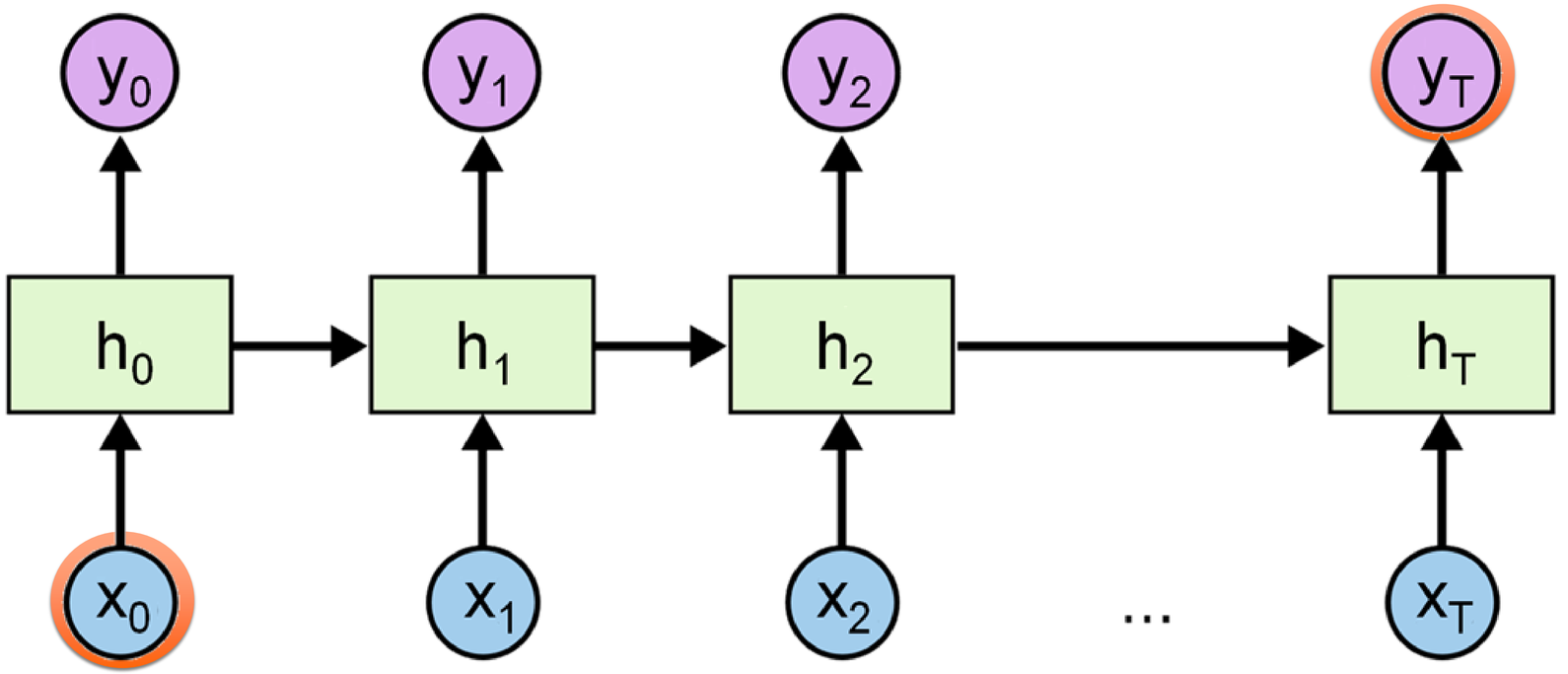
Image from Christopher Olah's blog

**prénom d'emprut*

Dépendances à long terme

Exemple

Qui est allé à Paris?



présenter sa recherche à une conférence d'apprentissage profond qui se déroulait à

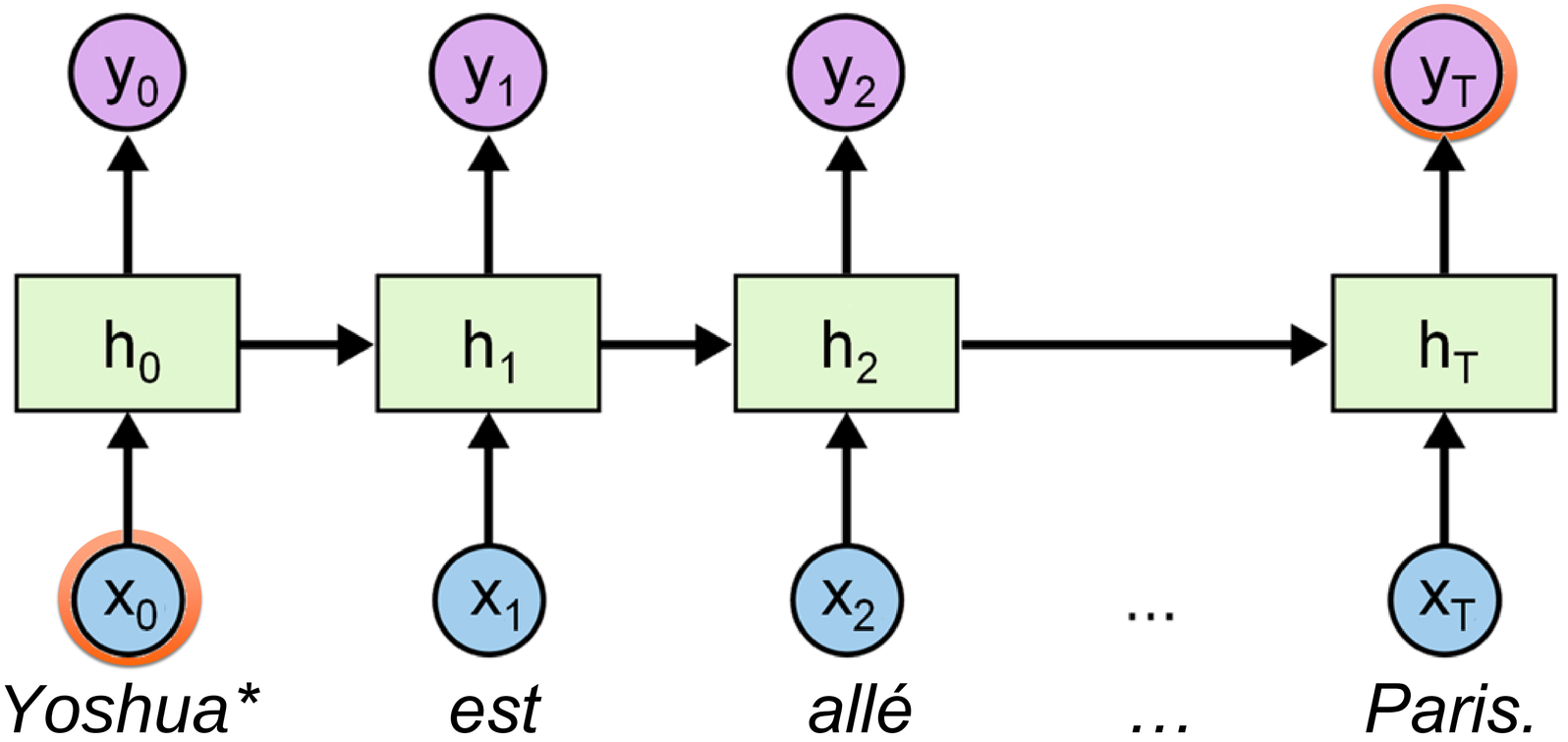
Image from Christopher Olah's blog

**prénom d'emprut*

Dépendances à long terme

Exemple

Qui est allé à Paris?



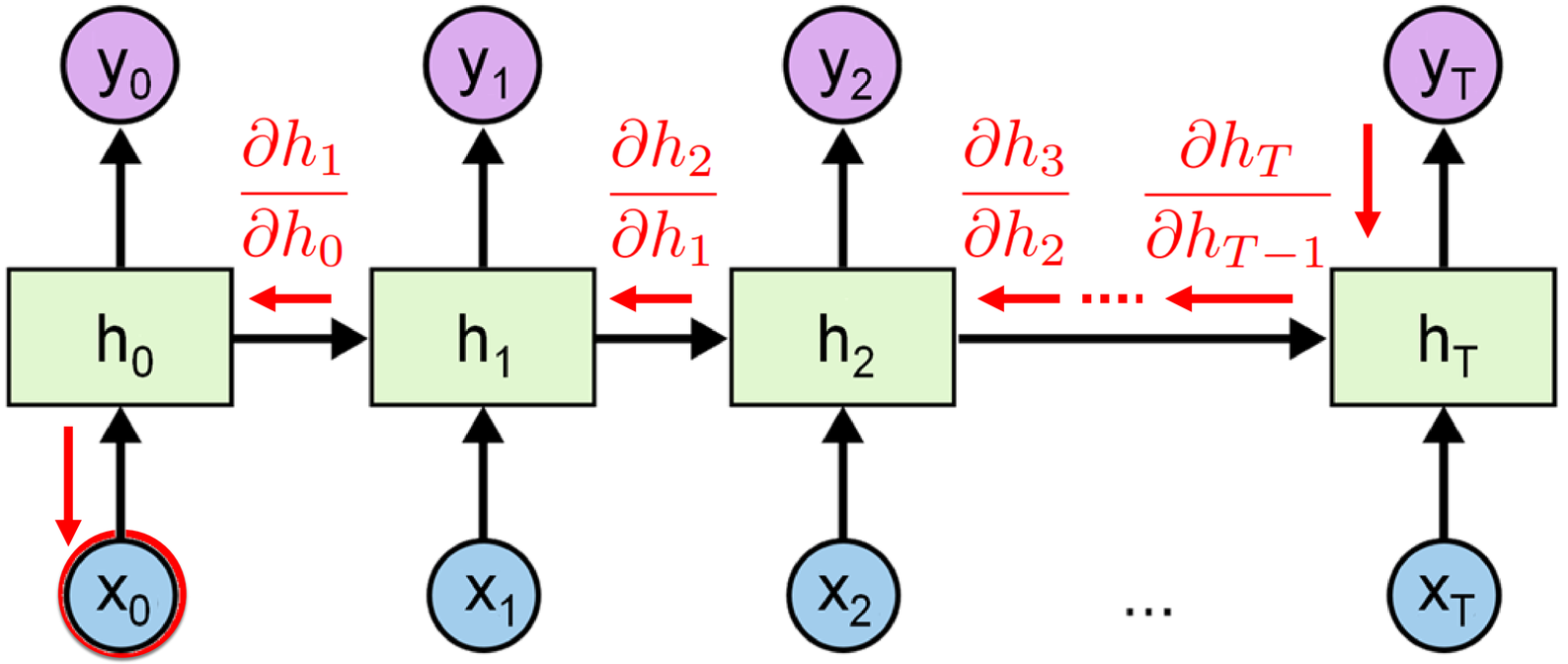
Un RNN doit être capable d'apprendre des dépendances à long terme!

Image from Christopher Olah's blog

**prénom d'emprut*

Dépendances à long terme

Propagation du gradient



Apprendre des dépendances à long terme

= Propager le gradient loin dans le temps

Image from Christopher Olah's blog

Difficultés d'apprentissage

Problème

- Pour apprendre des dépendances à long terme, il faut pouvoir propager le gradient loin dans le temps.

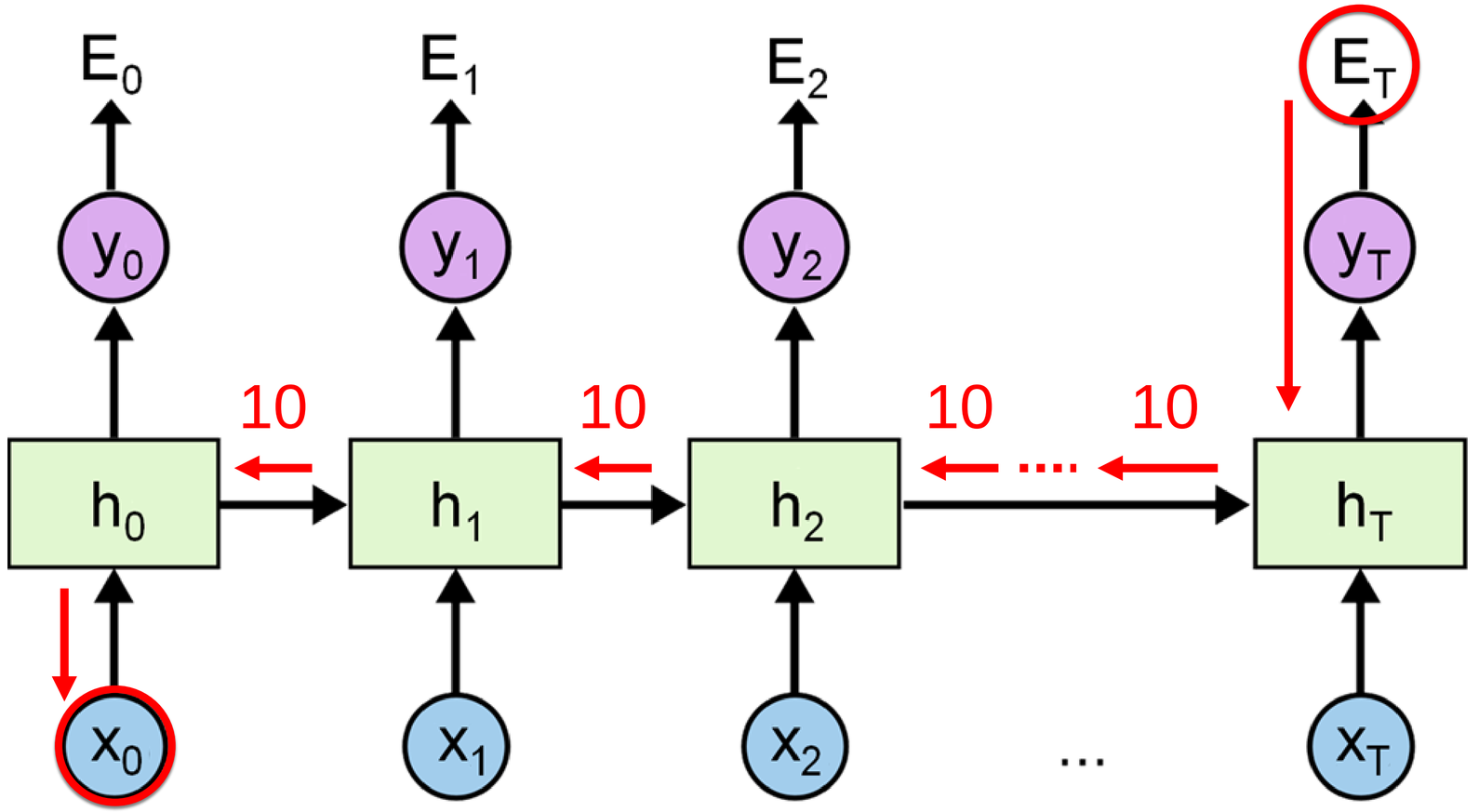
- La propagation des gradients à travers de nombreux pas de temps peut devenir instable et créer des problèmes d'entraînement.

  - $$\frac{\partial y_T}{\partial x_0} = \frac{\partial y_T}{\partial h_T}\frac{\partial h_T}{\partial h_{T-1}} \cdots \frac{\partial h_3}{\partial h_2}\frac{\partial h_2}{\partial h_1}\frac{\partial h_1}{\partial h_0}\frac{\partial h_0}{\partial x_0}$$

  - C'est la principale difficulté lors de l'entraînement de RNNs!

Explosion de gradient

Schéma



Gradient amplifié à chaque temps = Explosion!

Explosion de gradient

Solution

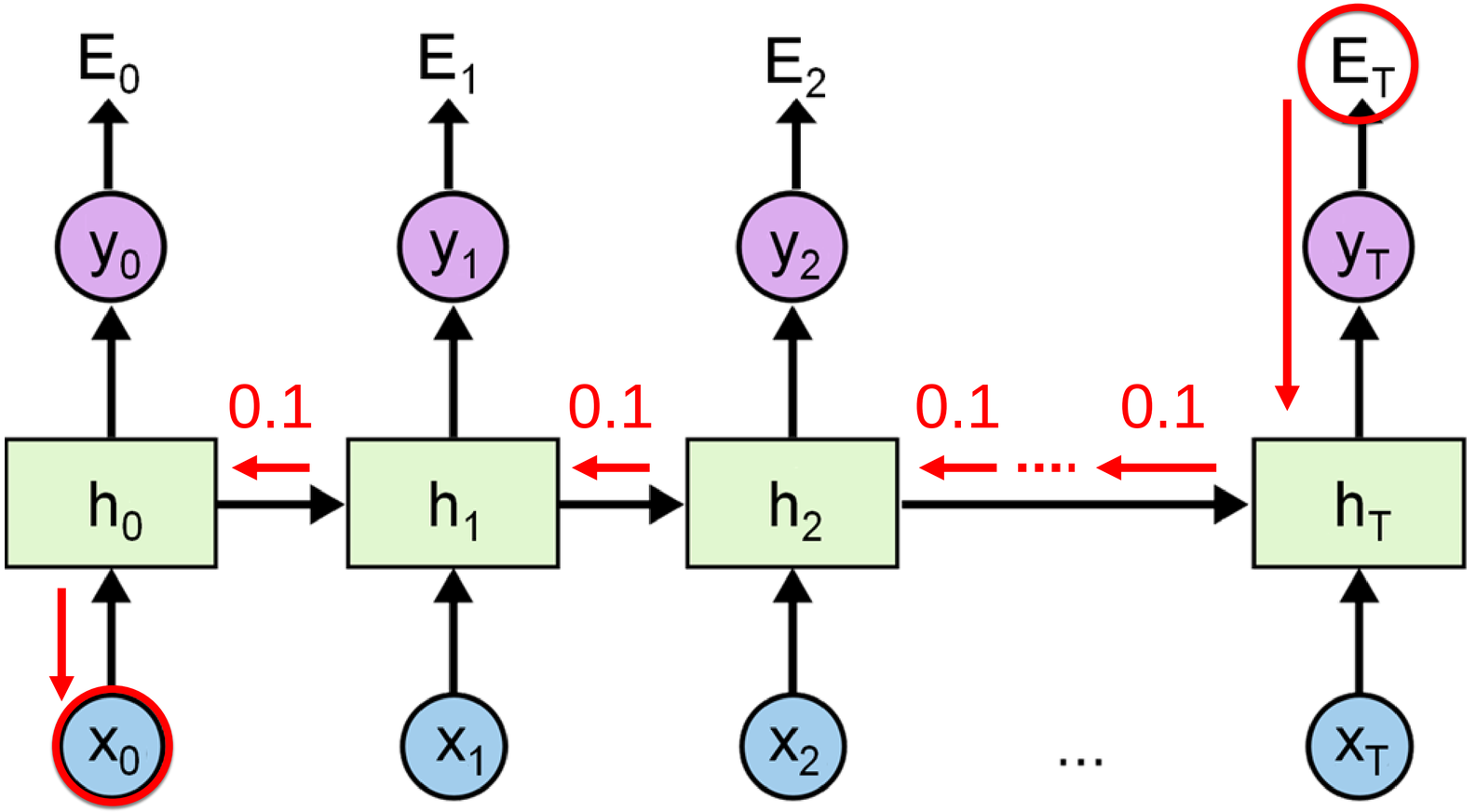
- Gradient amplifié à chaque temps → Explosion de Gradient!
  - Problème: Fait diverger les paramètres
- Solution simple: **Gradient Clipping**:

$$g = \frac{\partial E}{\partial W}$$
$$\textbf{if } ||g|| \geq \text{seuil } \textbf{then}$$
$$g \leftarrow \frac{\text{seuil}}{||g||} g$$
$$\textbf{end if}$$

Pascanu et al., *On the difficulty of training Recurrent Neural Networks*, 2013

Dissipation de gradient

Schéma

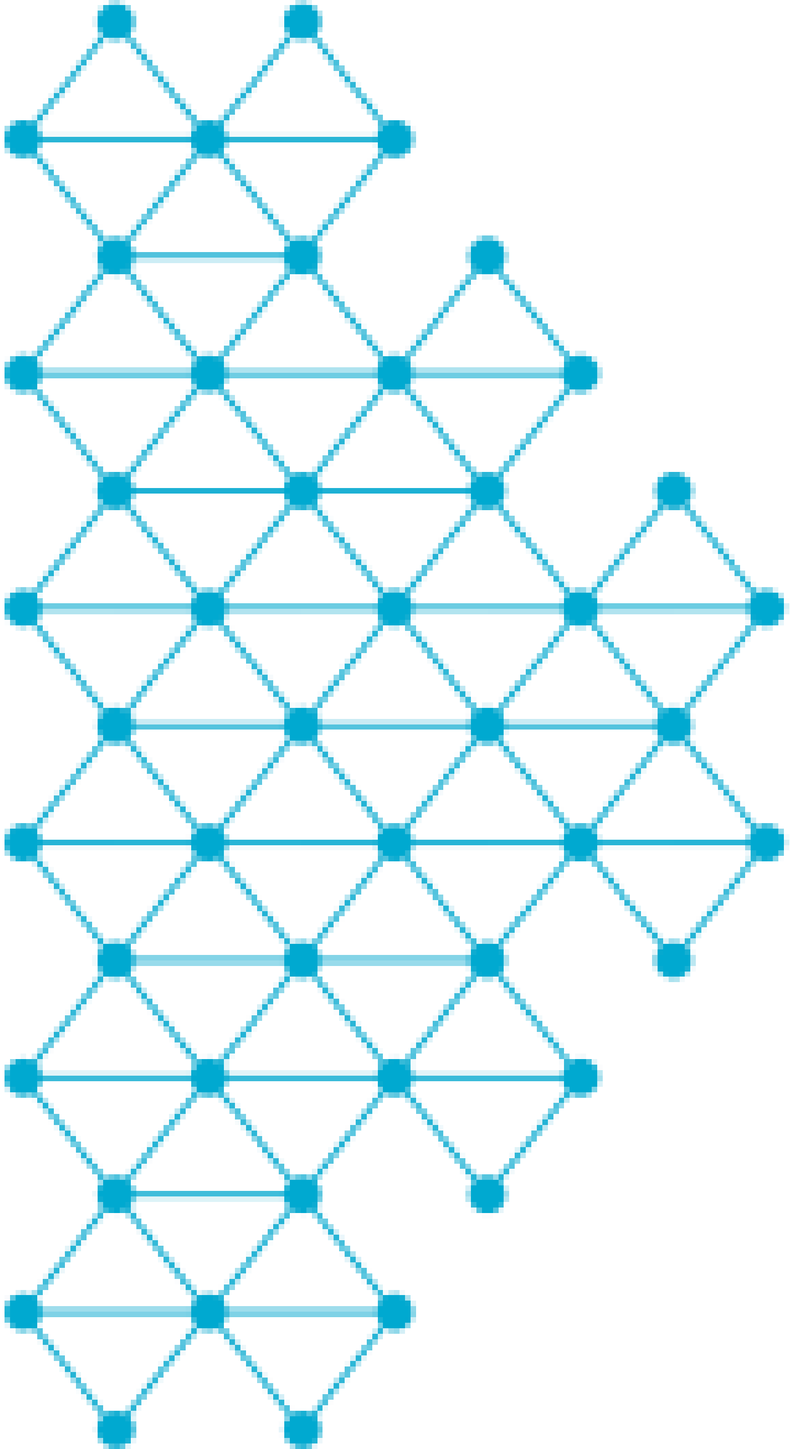


Gradient atténué à chaque temps = Dissipation!

Dissipation de gradient

Solution

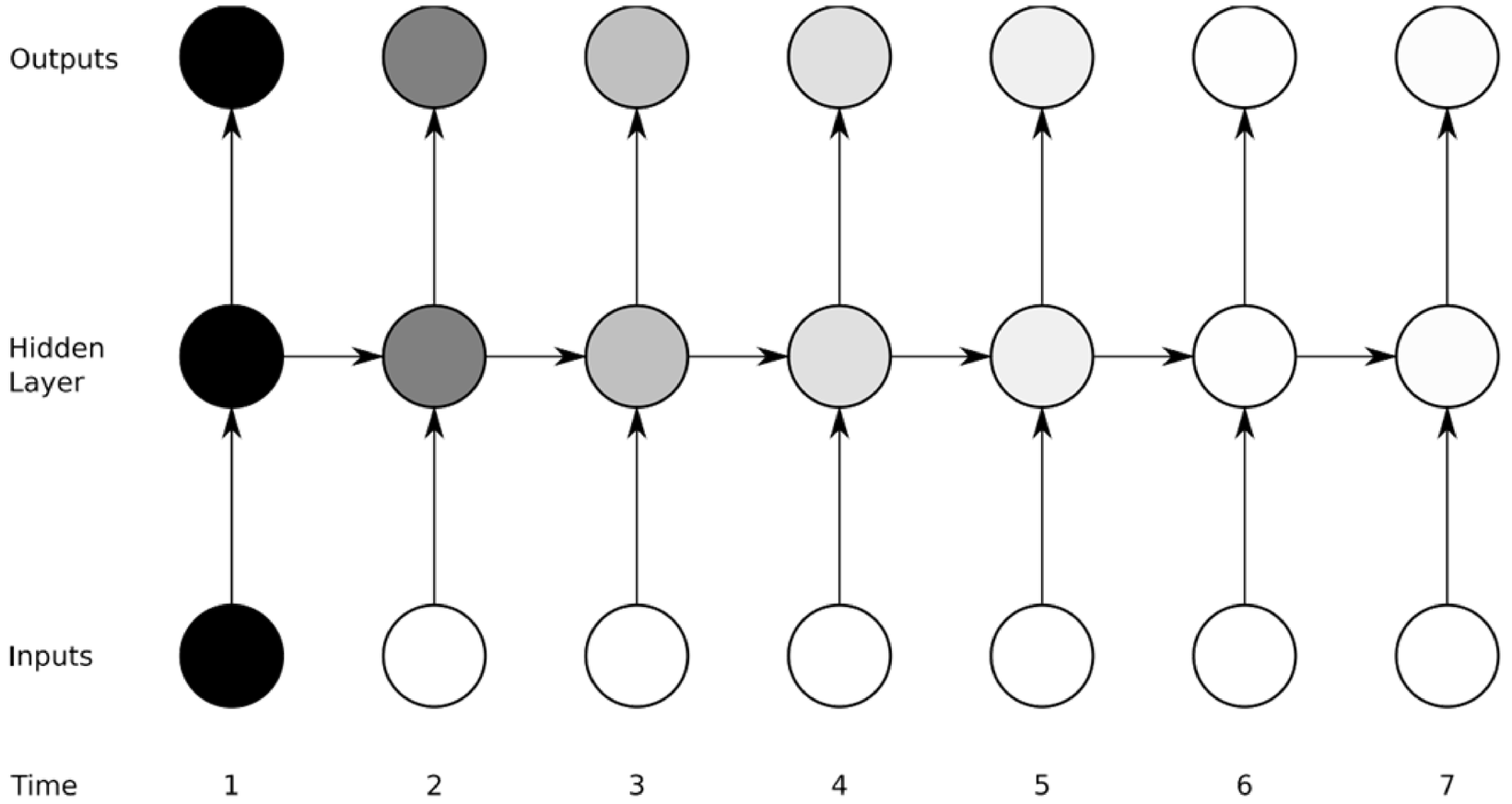
- Gradient atténué à chaque temps → Dissipation de Gradient!
  - Problème: Empêche l'apprentissage de dépendances à long terme!
- Pas de solution simple.
  - Utilisation d'architectures de RNN particulières.



5. Architectures de RNNs

Manque de mémoire

Présentation

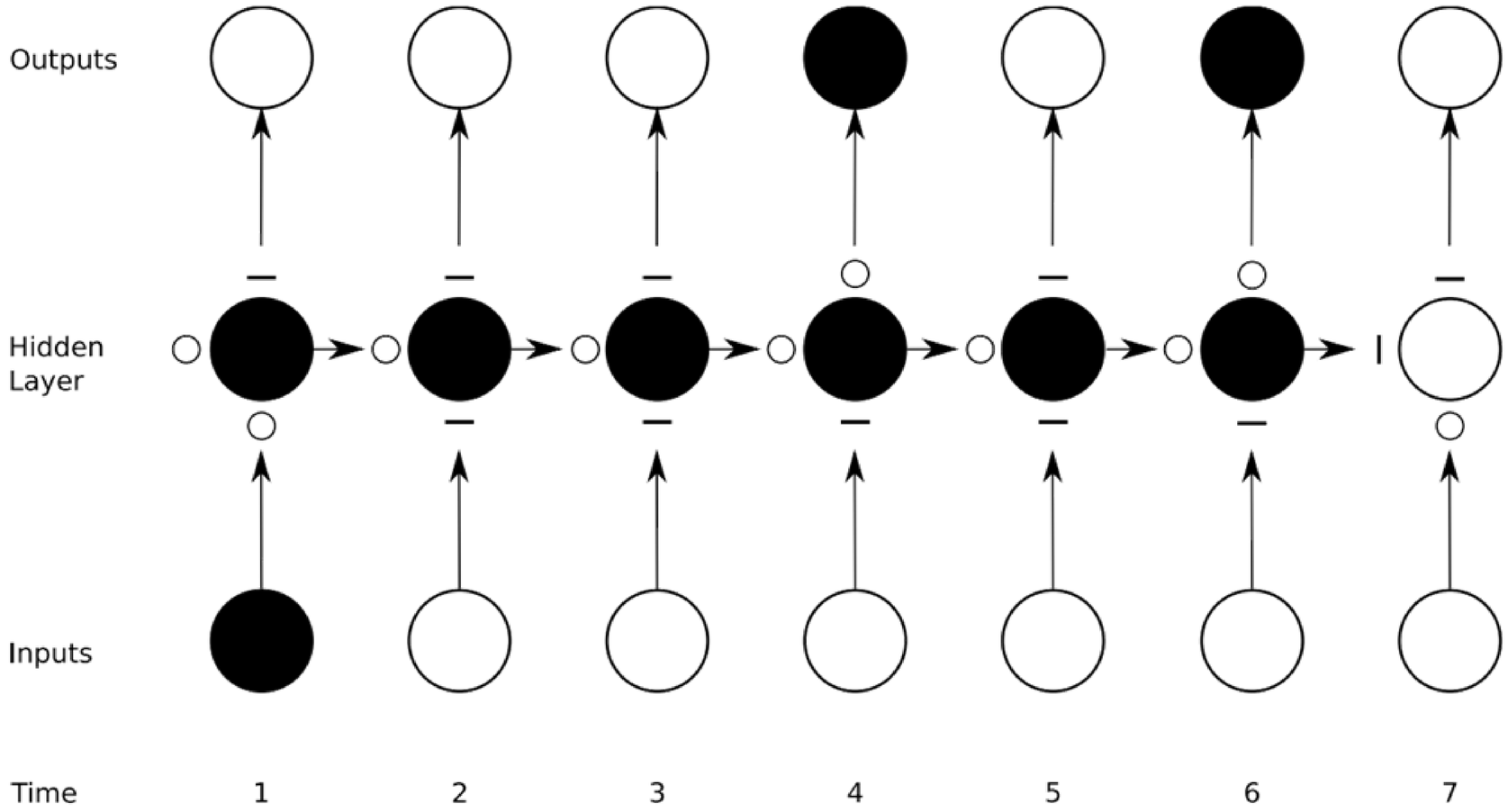


La nuance de gris montre l'influence de l'entrée du RNN au temps 1. Elle décroit au cours du temps, comme le RNN oublie peu à peu sa première entrée.

Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*

Manque de mémoire

Ajout de gates



En ajoutant 3 gates (o ouvert; - fermé), qui contrôlent l'entrée, la sortie et l'effacement de l'état, on peut retenir et propager l'information dans le temps.

Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*

Long Short-Term Memory (LSTM)

Introduction

Réduction du probème de disspation avec **un méchanisme de gates** et une **cellule mémoire**.

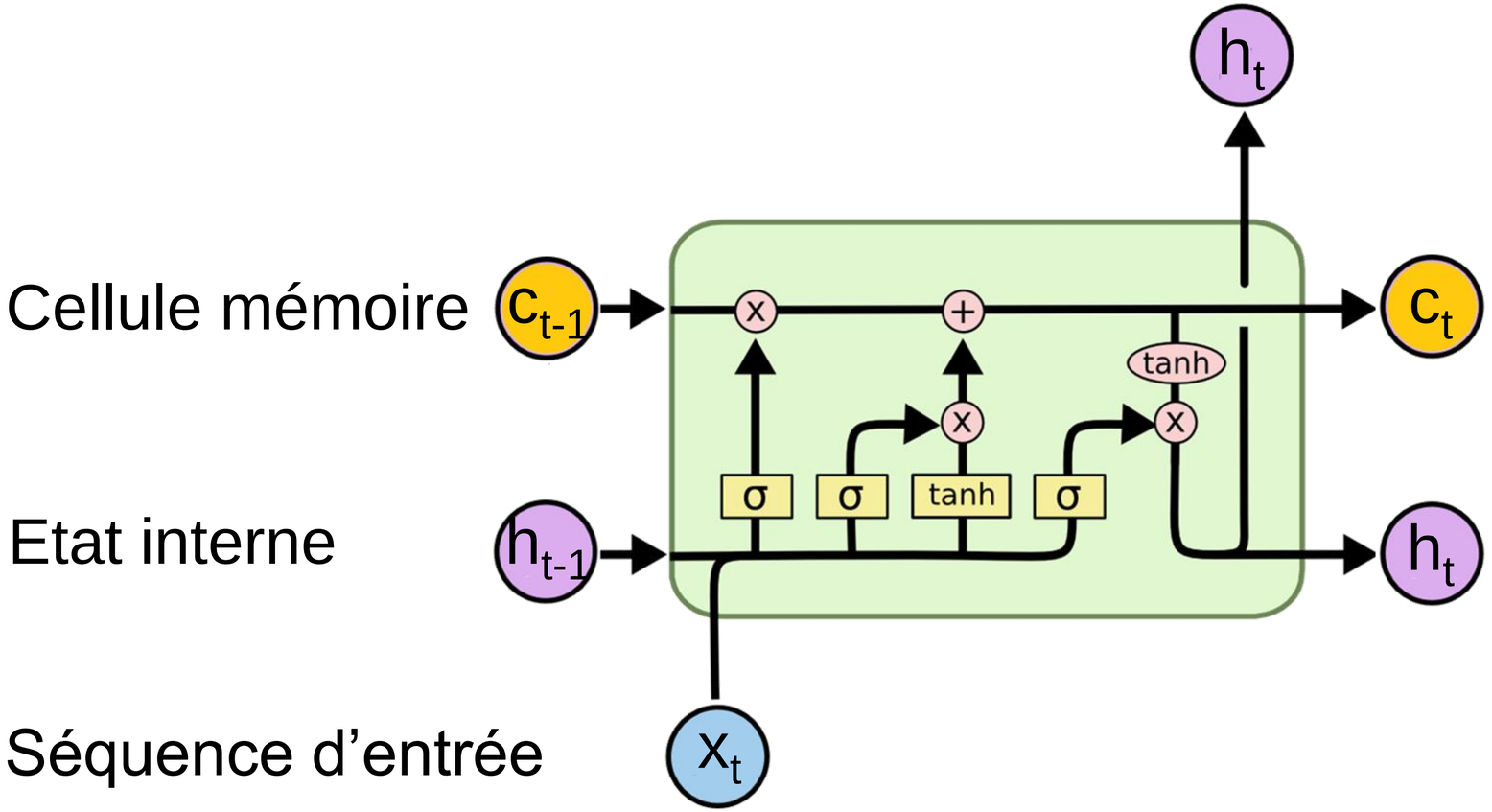


Image from Christopher Olah's blog

Hochreiter et al., *Long short-term memory*, Neural Computation 1997

LSTM pas à pas

La cellule mémoire

- Le point clé de la LSTM est sa cellule mémoire.
  - Très peu d'opérations dessus.
  - L'information peut passer très facilement.

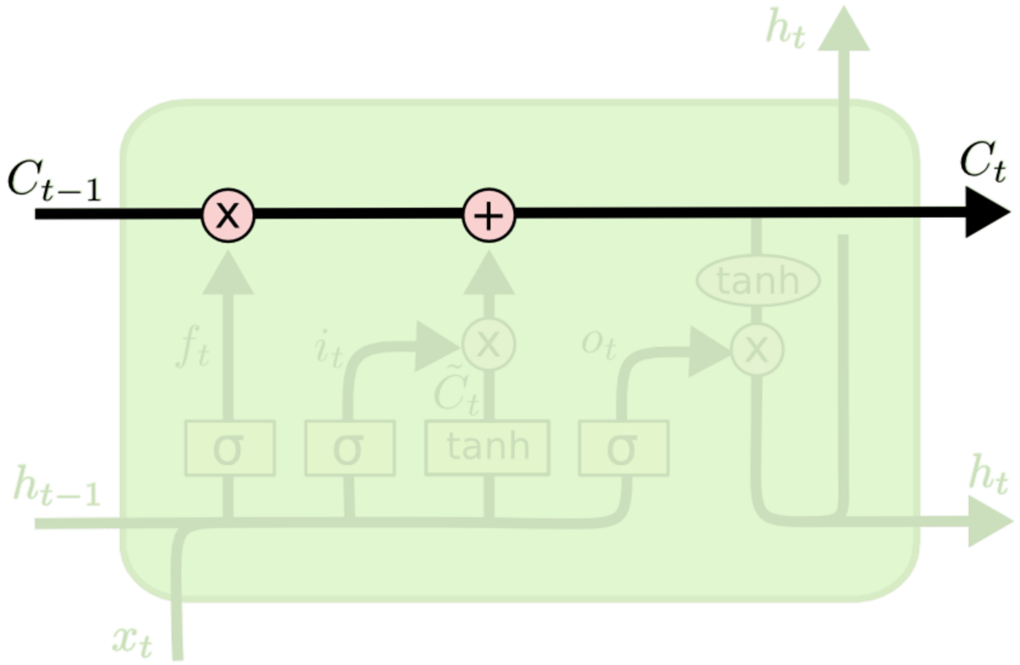
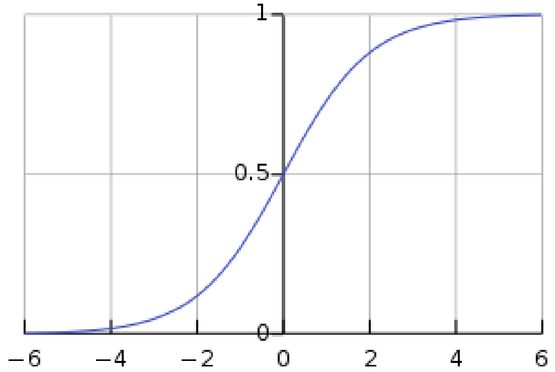


Image from Christopher Olah's blog

LSTM pas à pas

Calcul de la forget gate

- Calcul de la forget gate à partir de x_t et h_{t-1}.
  - $f_t = \sigma(U_f x_t + W_f h_{t-1} + b_f)$
  - σ est la fonction sigmoïde (bornée entre 0 et 1).



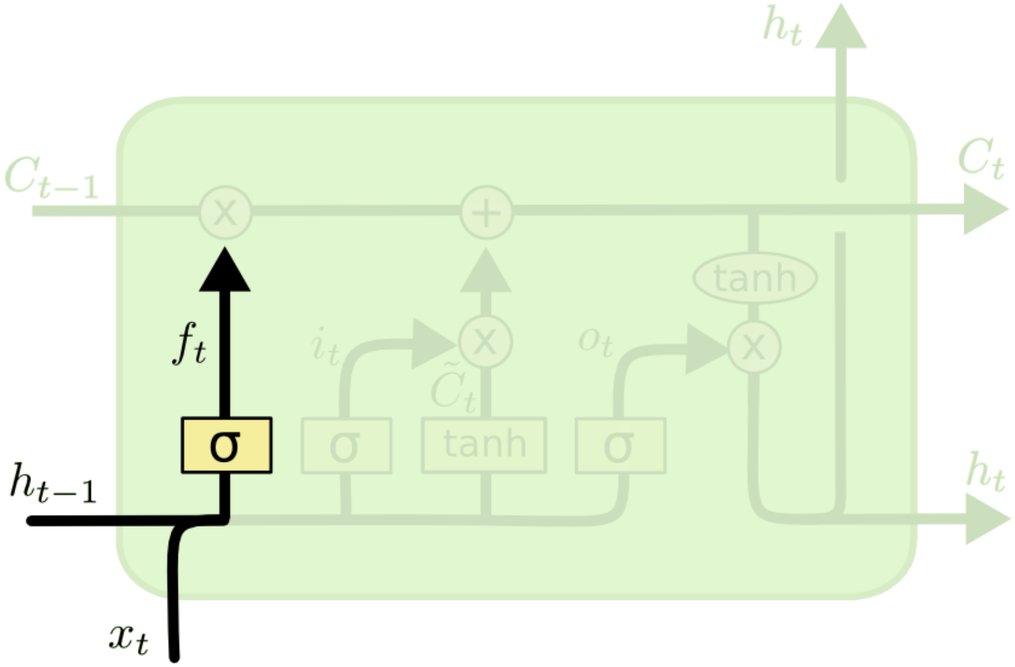


Image from Christopher Olah's blog

LSTM pas à pas

Calcul de la forget gate (2)

- Calcul de la forget gate à partir de x_t et h_{t-1}.
  - $f_t = \sigma(U_f x_t + W_f h_{t-1} + b_f)$
  - σ est la fonction sigmoïde
  - La forget gate permet d'oublier ce qui est présent dans la cellule mémoire.

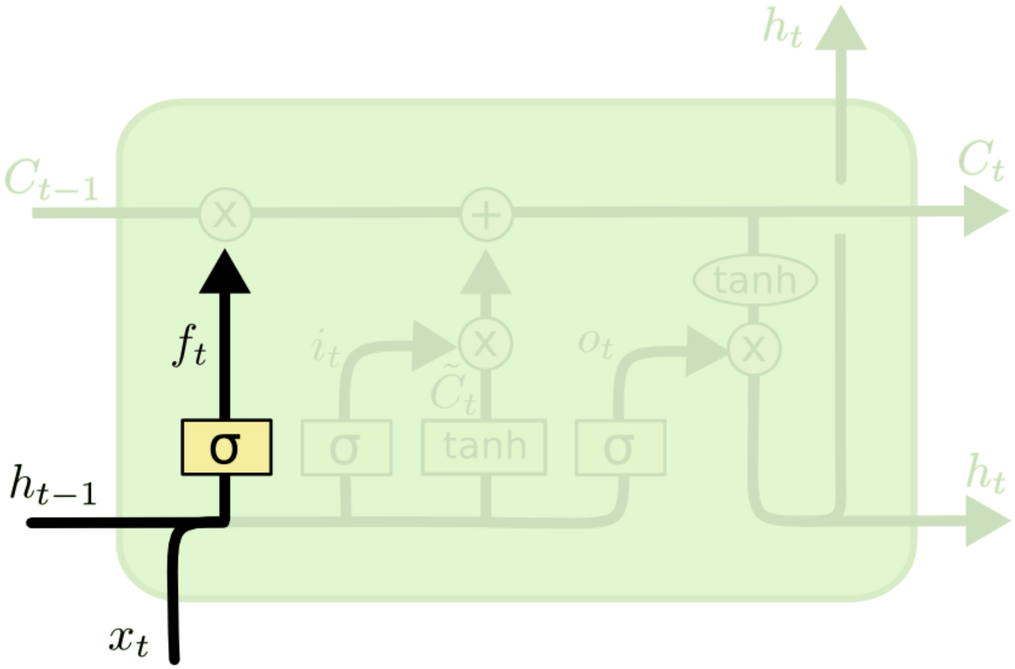


Image from Christopher Olah's blog

LSTM pas à pas

Calcul de l'input gate et du cell candidate

- Calcul de l'input gate à partir de x_t et h_{t-1}.
  - $i_t = \sigma(U_i x_t + W_i h_{t-1} + b_i)$
  - L'input gate contrôle ce qui entre dans la cellule mémoire.
- Calcul de ce qui va être ajouté à la cellule mémoire.
  - $g_t = \tanh(U_g x_t + W_g h_{t-1} + b_g)$

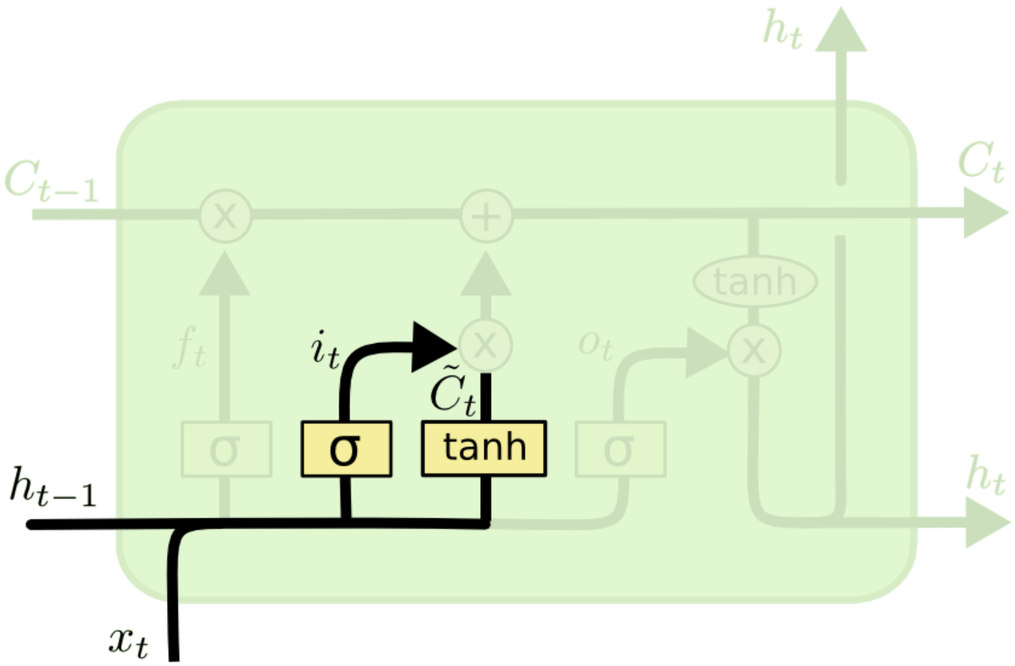


Image from Christopher Olah's blog

LSTM pas à pas

Calcul de la cellule mémoire

- Mise à jour de la cellule mémoire à l'aide de l'input et de la forget gate.
  - $c_t = i_t \odot g_t + f_t \odot c_{t-1}$
  - $\odot$ = multiplication terme à terme.
  - L'input gate permet d'ajouter de l'information dans la cellule, et la forget gate permet d'oublier l'information déjà présente dans la cellule.

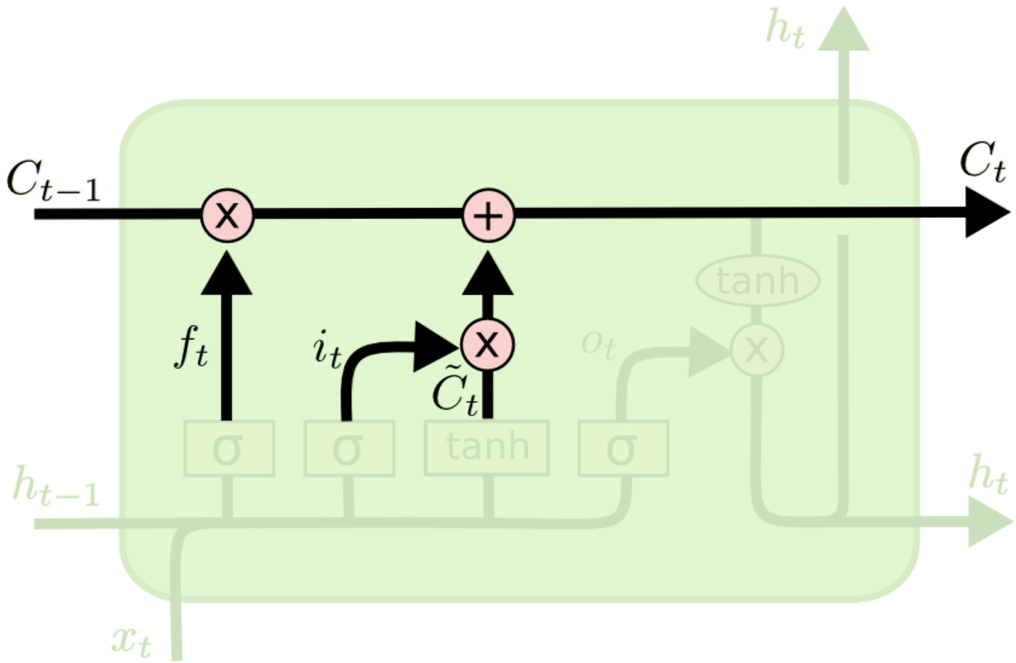


Image from Christopher Olah's blog

LSTM pas à pas

Calcul de l'output gate et de la séquence de sortie

- Calcul de l'output gate à partir de x_t et h_{t-1}.
  - $o_t = \sigma(U_o x_t + W_o h_{t-1} + b_o)$
  - L'output gate contrôle ce qui sort de la cellule mémoire.
- Calcul de l'état interne.
  - $h_t = o_t \odot \tanh(c_t)$

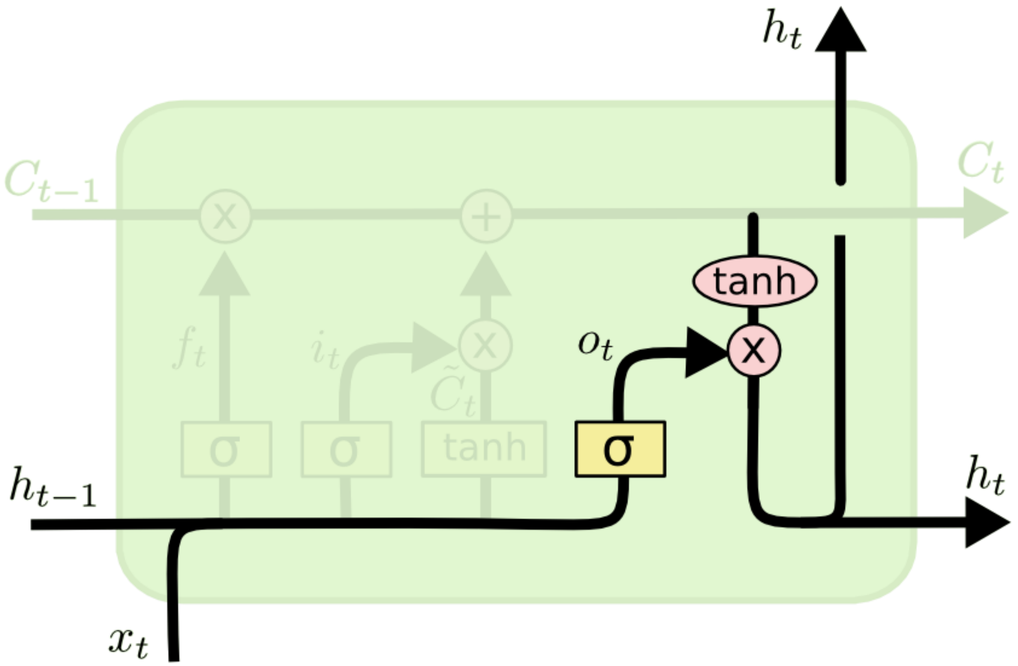


Image from Christopher Olah's blog

Long Short-Term Memory (LSTM)

En entier

Réduction du probème de disspation avec **un méchanisme de gates** et une **cellule mémoire**.

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

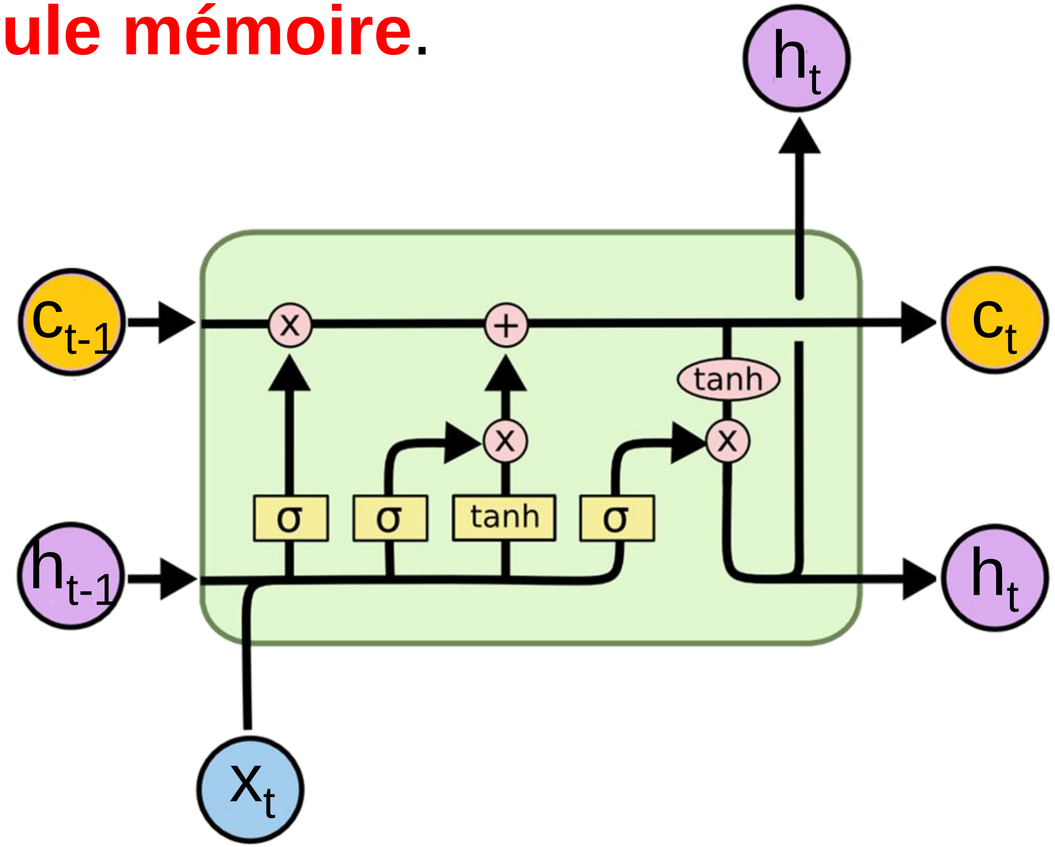


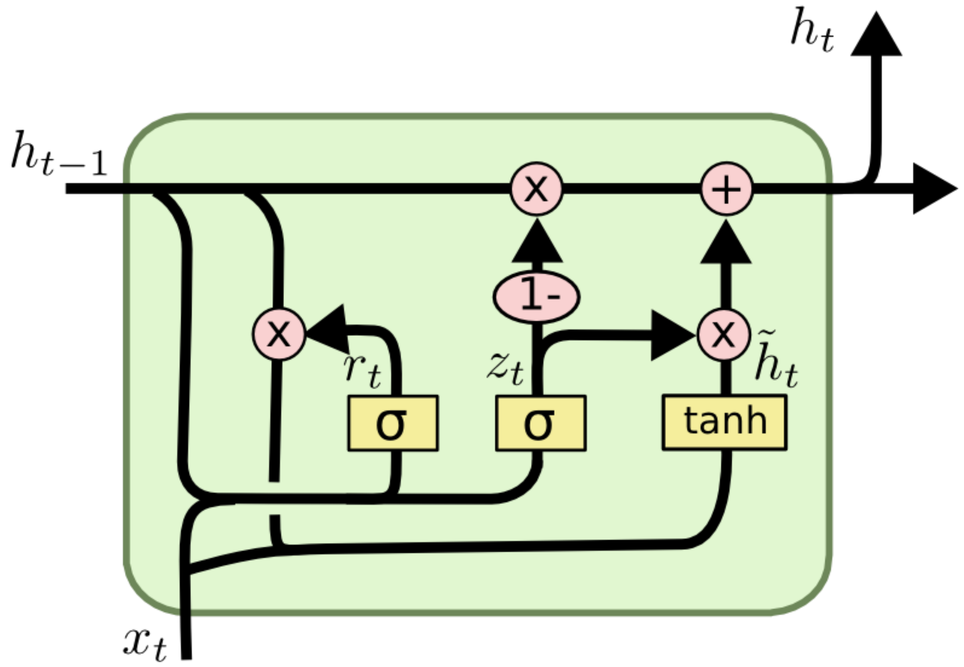
Image from Christopher Olah's blog

Hochreiter et al., *Long short-term memory*, Neural Computation 1997

Gated Recurrent Unit (GRU)

Aperçu

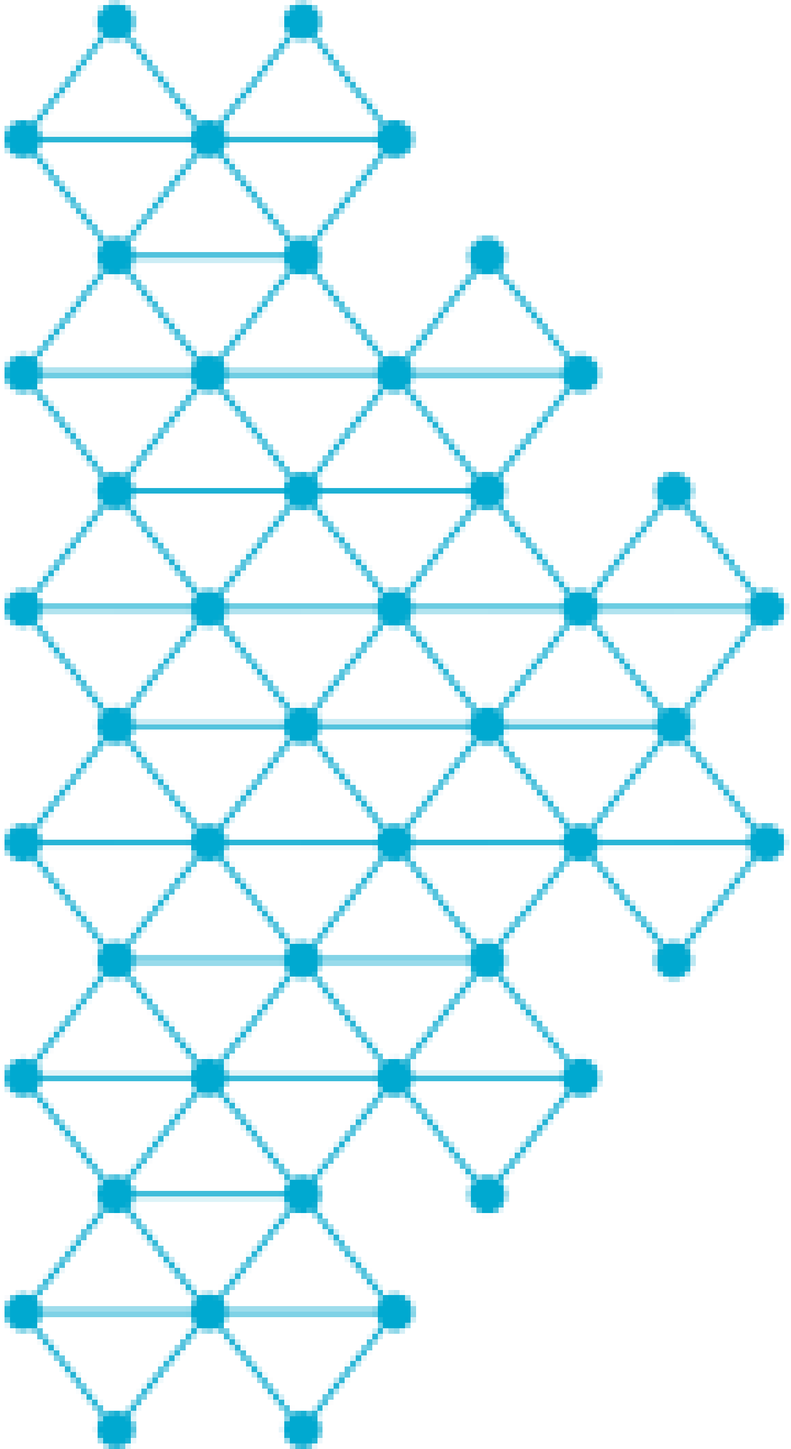
- Une variante populaire de la LSTM.
  - Pas de cellule mémoire explicite.
  - Input et forget gates combinées.
- En pratique, performances égales à la LSTM.
  - Plus rapide à calculer.



$$z_t = \sigma(U_z x_t + W_z h_{t-1} + b_z)$$
$$r_t = \sigma(U_r x_t + W_r h_{t-1} + b_r)$$
$$g_t = \tanh(U_g x_t + W_g(r_t \odot h_{t-1}) + b_g)$$
$$h_t = z_t \odot g_t + (1 - z_t) \odot h_{t-1}$$

Image from Christopher Olah's blog

Chung et al.: *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*



6. RNNs Génératifs

Exemple d’utilisation

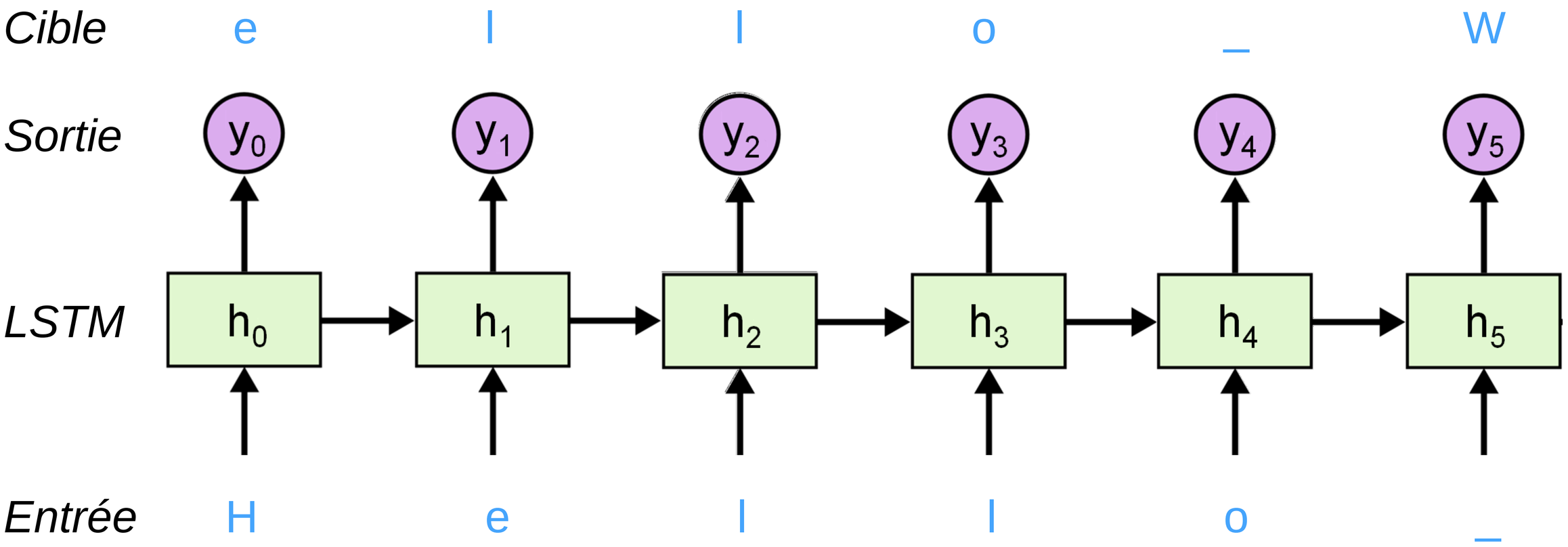
Définition de la tâche

But: Prédiction du prochain caractère de la séquence.

- Données: Texte (50 caractères: chiffres + lettres + espace + …)
- Cibles: $c_t = x_{t+1}$
- Erreur: Entropie croisée à chaque temps
- Modèle de langue

Exemple d'utilisation

Présentation du modèle



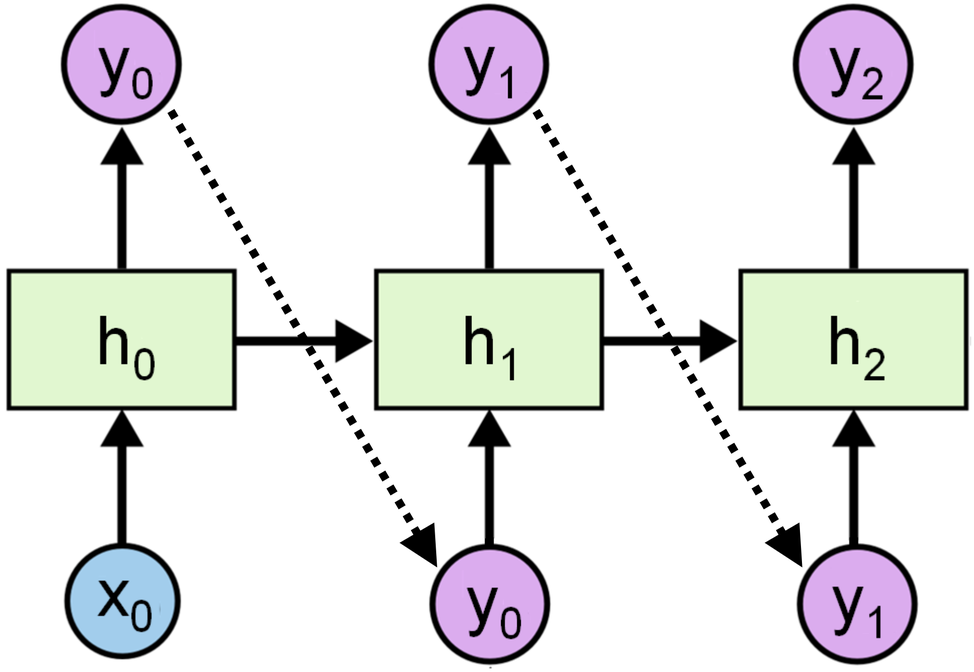
Erreur: Entropie croisée à chaque temps.

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!



Réseaux Récurrents Génératifs

Exemple de génération: Entraîné sur des textes de WSJ

Juste après l'initialisation:

- *« usb9xkrd9ruaias$dsaqj'4lmjwyd61\se.lcn6jey0pbco40ab'65<8um324 nqdhm<ufwt#y*/w5bt'nm.zq«2rqm-a2'2mst#u315w&tNwdqNafqh »*

Après la première époque:

- *« to will an apple for a N shares of the practeded 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 »*

Results from http://deeplearningathome.com/2016/10/Text-generation-using-deep-recurrent-neural-networks.html

Réseaux Récurrents Génératifs

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!

Tiré du blog d’Andrej Karpathy

Réseaux Récurrents Génératifs

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:

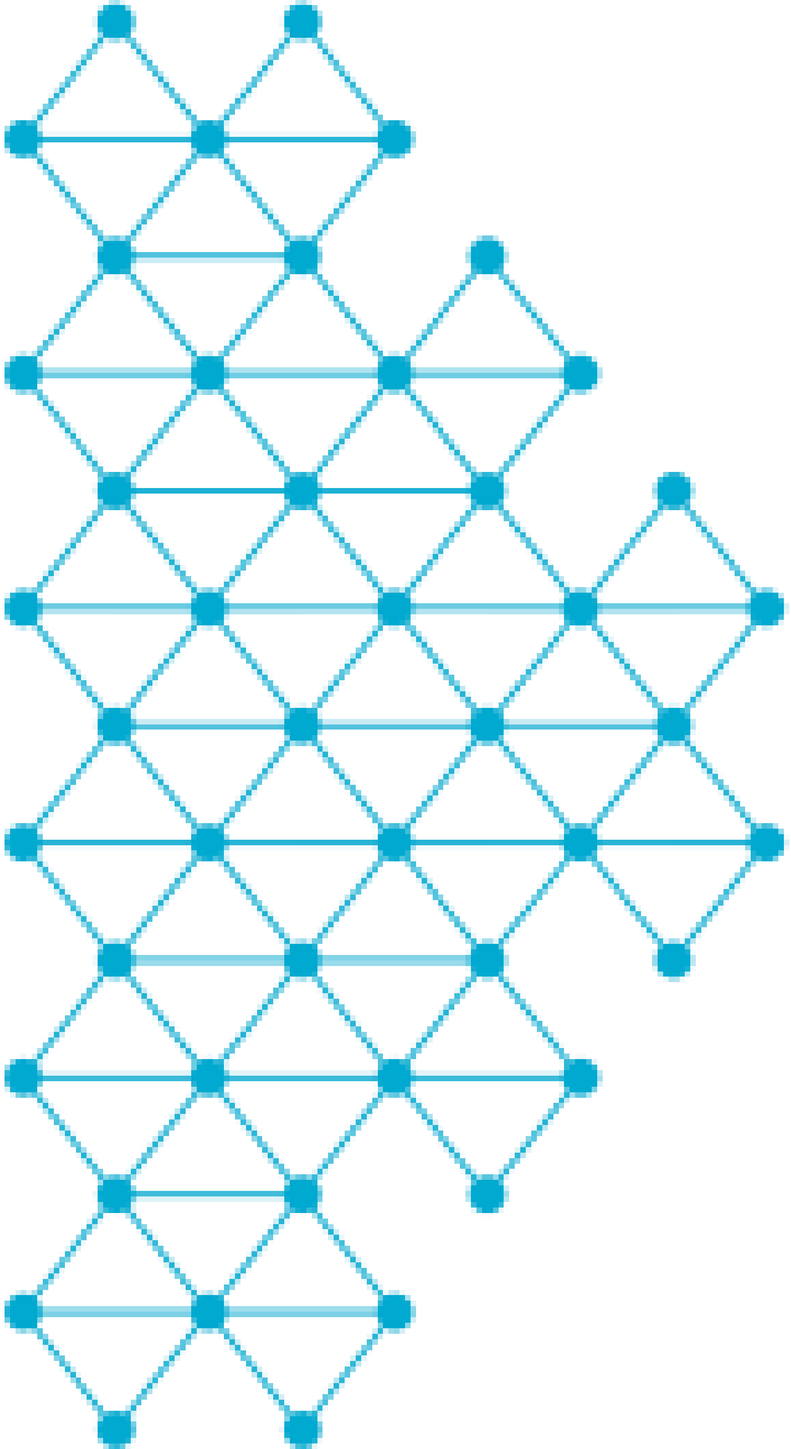
- Licence
- Imports
- Macros

Tiré du blog d’Andrej Karpathy

Réseaux Récurrents Génératifs

Exemple supplémentaires

- Nombreuses autres possibilités:
  - Génération de musique
  - Génération d’écriture manuscrite
  - Génération de parole



Questions ?

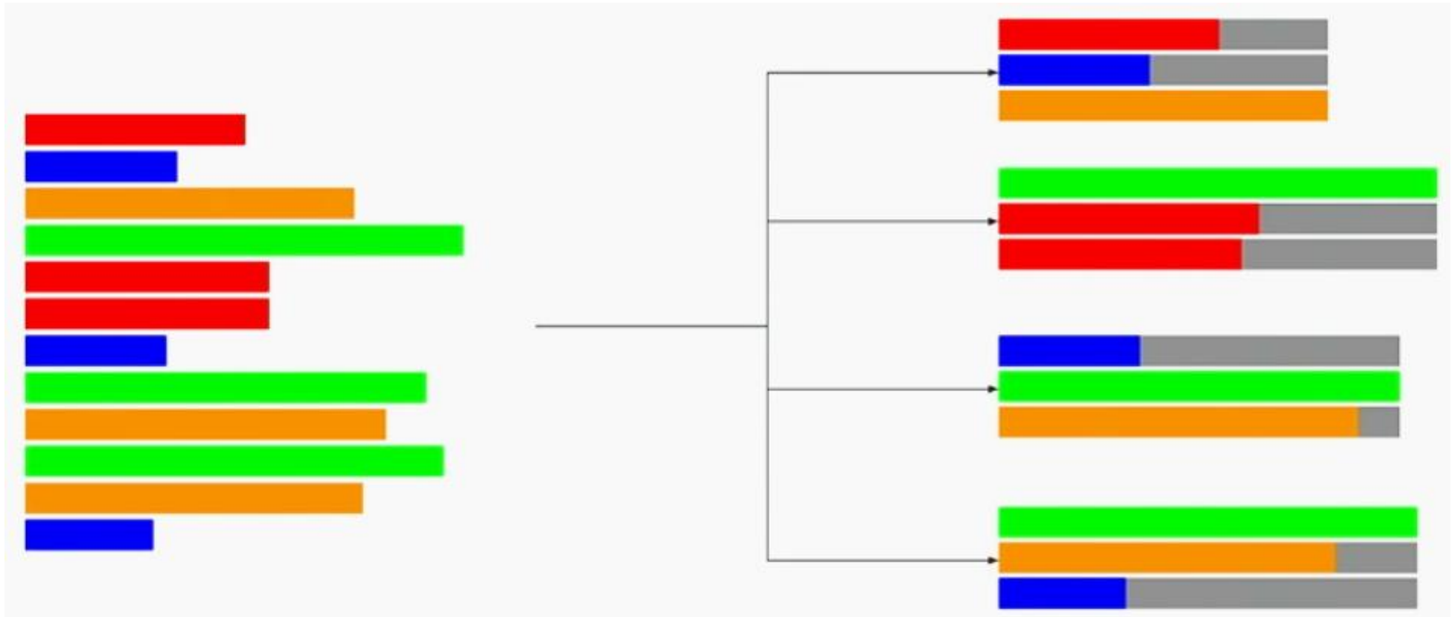
Rajout de zéros

Introduction

Comment faire des mini-lots (mini-batches) avec des séquences de taille différentes?

- Rajouter des zéros à la fin des séquences (zero-padding)!

Dataset



Mini-batches

Image from https://blog.altoros.com/the-magic-behind-google-translate-sequence-to-sequence-models-and-tensorflow.html

Rajout de zéros

Tailles très différentes

Comment faire des mini-lots (mini-batches) avec des séquences de taille différentes?

- Si les tailles sont très différentes, il peut être avantageux de trier les séquences!

Dataset

Mini-batches

Image from https://blog.altoros.com/the-magic-behind-google-translate-sequence-to-sequence-models-and-tensorflow.html