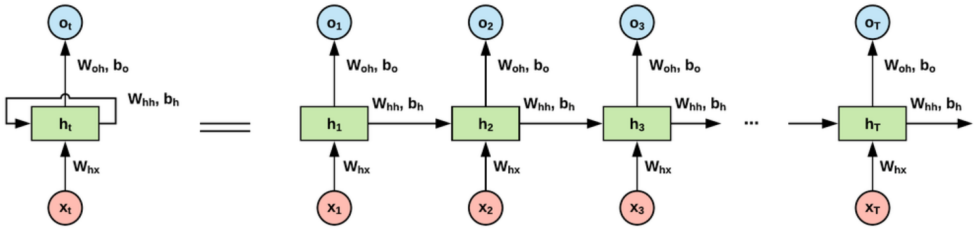
Les réseaux de neurones récurrents

Introduction

Les réseaux de neurones classiques comme les *fully-connected* ou les CNN ne disposent pas d'une mémoire résumant toute l'information identifiée autrement que par les poids du réseau. Cependant ces poids ont comme but de permettre d'effectuer de meilleurs prédictions, pas de garder de l'information des *n* précédents intrants.

L'idée des réseaux de neurones récurrents est donc de répondre à cette problématique en introduisant une cellule cachée *hidden state* permettant une information résumée des précédents intrants. Plus formellement, l'intrant sera considéré comme une suite (ou séquence) $(x_1^i, \ldots, x_t^i)$ et à chaque temps, chaque x_t^i sera propagé dans le réseau comme dans un fully-connected à la différence qu'on ajoutera le résultat d'une cellule h_{t-1} qui contiendra toute l'information sur les observations antérieures à x_t^i.



Source : Medium.com
Équations : $h_t = tanh(W_{hx}x_t + W_{hh}h_{t-1} + b_h)$ *et* $o_t = f(W_{oh}h_t + b_o)$

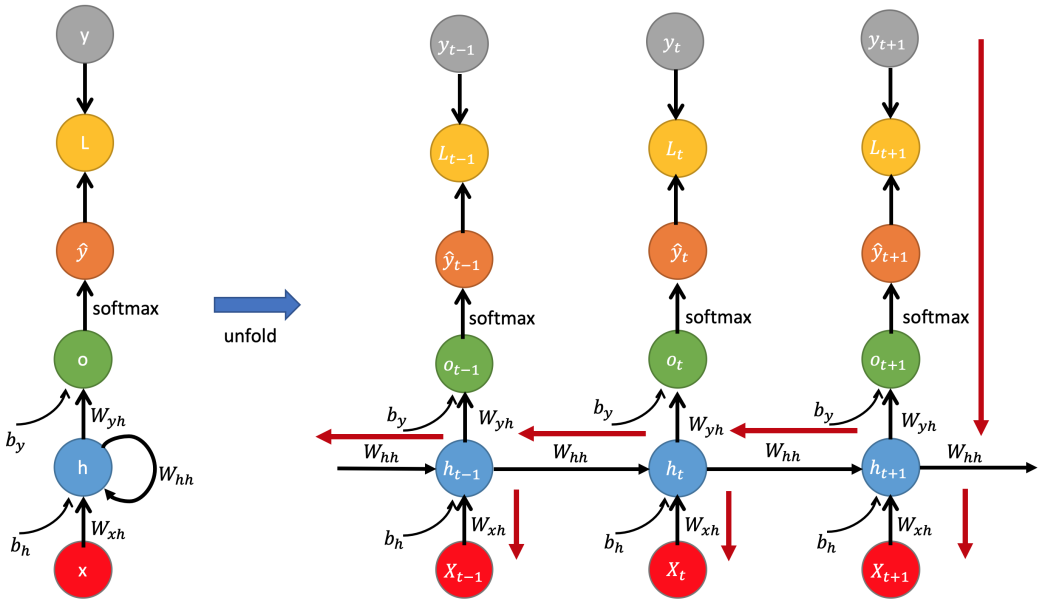
Dans l'exemple précédent (appelé le vanilla RNN), la sortie o_t est fonction de l'intrant x_t, des poids et biais et surtout de l'information condensée obtenue aux temps précédents h_t.

Quid de l'initialisation ? : Notons tout d'abord que comme dans un CNN, les paramètres sont partagés. Il en vient que les poids W*{hx}, W*{hh} et W_{oh} sont les mêmes à chaque temps et *a priori*, ils sont actualisés à chaque batch.

Cette démarche est impératif pour un grand nombre de problématiques telle que la compréhension de contexte, la prédiction de séries temporelles.

Un avantage de cette architecture est que les intrants peuvent avoir des tailles (par exemple, si les séquences sont des phrases alors il n'est pas nécessaire d'avoir des tailles fixes pour chaque phrase).

L'apprentissage est quasiment similaire à celui des CNN ou des fully-connected : il s'appuye sur l'algorithme de rétropropagation du gradient sur la forme déroulée du réseau de neurone récurrent. Néanmoins, il faut prendre en compte la dimension temporelle des données : sur l'illustration ci-après, il s'agit des flèches rouges entre les unités cachées (en bleu).



Source : https://mmuratarat.github.io/

Problème lors de l'apprentissage : si la séquence est longue alors l'apprentissage peut devenir difficile soit parce que le gradient disparaît (vanishing gradient) ou soit parce que le gradient explose (explosing gradient). En effet, dans les algorithmes de rétropropagation, une multiplication est effectuée d'une couche à l'autre provoquant une croissance exponentielle en fonction du nombre de couches remontées.