Comprendre le fonctionnement d'un LSTM

Introduction

En Deep Learning, un des réseau de neurone récurrent le plus utilisé pour prédire des séries temporelles est appelé LSTM, l'acronyme de Long Short-Term Memory".

On parle de mémoire à court et long terme comme analogie de la mémoire humaine car comme tout réseau de neurone récurrent, le LSTM est capable soit de prendre en compte les données des couches précédentes mais aussi d'oublier de l'information car devenu moins utile.

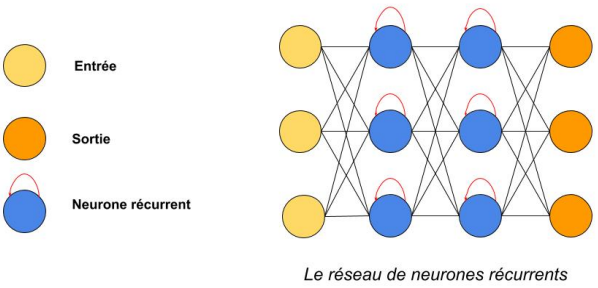
Le problème des réseaux de neurones récurrents (RNN)

Comment fonctionne un RNN ?

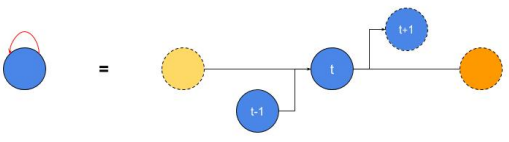
Voici à présent le schéma d'un RNN:

Contrairement à un ANN (et c'est là la seule différence), sur chaque neurone bleu, on a une boucle (développée sous le schéma du réseau): on a une donnée t (3 valeurs, une pour chaque neuron jaune) qui arrive dans le réseau:

- celles-ci *se propagent* dans le réseau comme dans un ANN
- sauf que chaque neurone bleu, en plus de recevoir les sorties pondérées des neurones précédents, il reçoit également *la valeur qui sortait de lui-même pour la donnée t -1*
- la valeur de sortie de chaque neurone bleu est conservée et servira pour la donnée *t + 1*



Le réseau de neurones récurrents



On a donc une **mémoire d'une itération** pour les neurones bleus.

Un problème d'apprentissage et de mémoire

Comme pour le perceptron en son temps, le RNN souffre du problème de dissipation du gradient. Qu'est-ce que cela signifie ?

Pour « apprendre », un RNN utilise la méthode de la descente du gradient afin de mettre à jour les poids entre ses neurones.

Cela repose sur la formule suivante :

$$w := w - \alpha \cdot Fw$$

avec :

w: $un\ poids\ du\ réseau$
α: $la\ vitesse\ d'apprentissage\ du\ réseau$
$Fw\ le\ gradient\ du\ réseau\ par\ rapport\ au\ poids\ w$

Problème

La mise à jour des poids se fait de droite à gauche, à mesure que l'on avance vers la gauche, le produit $\alpha \cdot Fw$ devient très petit et les poids des premières couches de neurones ne sont quasiment pas modifiés !

Ainsi, ces couches n'apprennent strictement rien…
Par conséquent, le RNN peut facilement oublier des données un petit peu anciennes (ou des mots assez éloignés du mot courant dans un texte) lors de la phase d'apprentissage : **sa mémoire est courte**.

Vers une meilleure architecture : le LSTM

Si une cellule (neurone) d'un RNN est finalement très simple (on concatène deux vecteurs puis on applique tanh dessus), ce n'est pas le cas du LSTM et du GRU au premier abord.

Les LSTM et GRU ont été créés comme méthode permettant de gérer efficacement la mémoire à court et long terme grâce à leurs systèmes de portes.

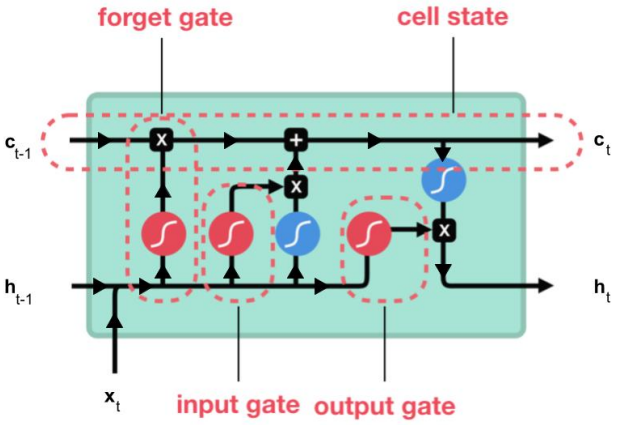
S'il en existe de nombreuses variantes, les versions d'origine (présentées ici) sont encore très très largement utilisées dans les meilleurs modèles de deep learning pour le traitement automatique du langage naturel, ce qui a trait à la reconnaissance/synthèse vocale mais aussi pour la génération de texte ou l'étude de marchés…

Comment fonctionne le LSTM

LSTM, qui signifie Long Short-Term Memory, est une cellule composée de trois « portes »: ce sont des zones de calculs qui régulent le flot d'informations (en réalisant des actions spécifiques).

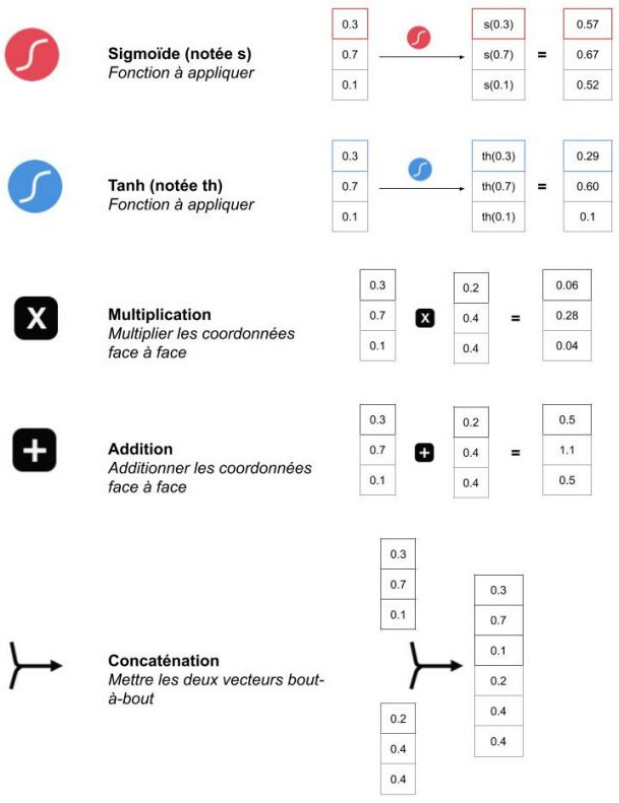
On a également deux types de sorties (nommées états).

- Forget gate (porte d'oubli)
- Input gate (porte d'entrée)
- Output gate (porte de sortie)
- Hidden state (état caché)
- Cell state (état de la cellule)

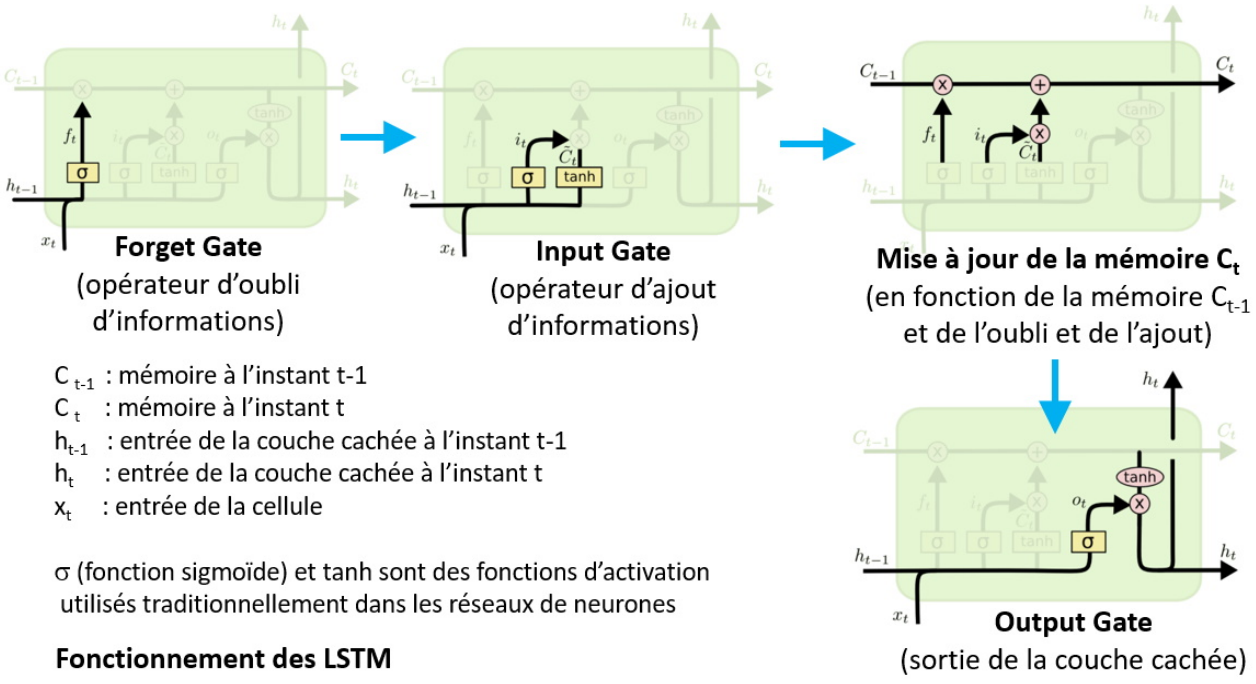


Cellule LSTM (crédit : image modifiée de Michaël Nguyen)

Commençons par comprendre toutes les opérations possibles dans un LSTM pour ne pas buter dessus ensuite.



Comment fait le réseau LSTM pour apprendre ?



Principe de fonctionnement d'une cellule LSTM d'un réseau récurrent

Ces opérations dans les portes permettent au LSTM de conserver ou supprimer des informations qu'il a en mémoire.

Par exemple, dans notre phrase « Hier soir j'ai mangé un hamburger et des », il est important de retenir les mots « hamburger » et « manger » tandis que les déterminants « un », « et » peuvent être oubliés par le réseau.

Les données stockées dans la mémoire du réseau sont en fait un vecteur noté ct: l'état de la cellule. Comme cet état dépend de l'état précédent $ct - 1$, qui lui-même dépend d'états encore précédents, le réseau peut conserver des informations qu'il a vu longtemps auparavant (contrairement au RNN classique).

Références

Comprendre le fonctionnement d'un LSTM et d'un GRU en schémas (Par **Lambert R.** - 9 octobre 2019), Original post: "Illustrated Guide to LSTM's and GRU's: A step by step explanation" (Par **Michael P. -** Sep 24, 2018)

Les réseaux de neurones récurrents pour les séries temporelles (Par **Patrick H. -** 6 septembre 2021)