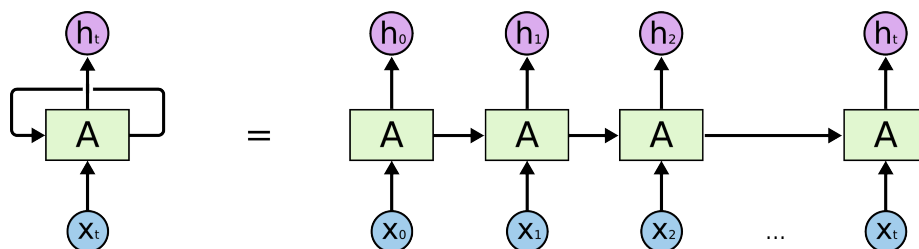


TD : **keras** & réseaux de neurones récurrents

Romain Tavenard

Les réseaux de neurones récurrents (*Recurrent Neural Networks*, RNN) sont des réseaux particuliers qui permettent de prendre en compte la notion de séquence d'observations.

1 Les RNN en (très) bref



L'illustration ci-dessous est issue du (très bon) blog de Christopher Olah et plus précisément d'un *post* dédié aux neurones LSTM (un cas particulier de neurone récurrent) disponible [ici](#). Cette illustration représente une couche récurrente d'un réseau de neurone. On voit sur la partie de gauche de cette figure que cette couche prend en entrée une observation x_t (où t est l'indice de l'observation dans la séquence) et retourne un vecteur h_t . On remarque surtout, ce qui n'était pas le cas pour les couches que vous avez étudiées jusqu'à présent, qu'il existe une boucle de rétroaction.

Pour comprendre le fonctionnement de cette boucle, il faut jeter un oeil à la partie droite de l'illustration, dans laquelle on a "déplié" la couche récurrente. Il faut comprendre ici que "déplié" signifie que l'on a explicité son fonctionnement pour le traitement de toute la séquence d'observations de x_0 à x_t . On peut alors remarquer que pour générer h_1 , la couche récurrente va non seulement regarder le contenu de x_1 , mais également obtenir de l'information concernant les états passés (par le biais de la flèche allant de la boîte A du temps 0 à celle du temps 1). C'est là le principe de base des couches récurrentes : elles encodent la connaissance acquise grâce aux observations passées et la prennent en compte en plus de l'observation courante pour fournir une sortie h_t qui soit représentative du contenu global de la séquence et non pas seulement de sa t -ième observation.

Les détails mathématiques de cet encodage varient selon les types de couches récurrentes. On peut citer ici les couches suivantes :

- RNN standard (appelées **SimpleRNN** en **keras**) ;
- LSTM, ou *Long Short-Term Memory* (**LSTM** en **keras**) ;
- GRU, ou *Gated Recurrent Unit* (**GRU** en **keras**).

Pour obtenir plus d'informations sur ces variantes, je vous renvoie vers le *post* de blog de Christopher Olah cité en début d'énoncé.

1.1 En pratique dans keras

Pour utiliser ce type de modèles dans **keras**, il faudra :

- se pencher sur la définition de couches récurrentes [doc] ;
- prendre garde à la dimension des données fournies en entrée au modèle car une couche récurrente s'attend à recevoir des données de dimension (sz, d) où sz est la taille maximale des séquences fournies en entrée et d est la dimension de chaque observation dans la séquence.

2 Énoncé du TD (correction disponible sur CURSUS)

1. Téléchargez les représentations **word2vec** disponibles à l'adresse : <http://nlp.stanford.edu/data/glove.6B.zip> (pour la langue anglaise). Téléchargez également le corpus textuel *WebKB* constitué d'un jeu d'apprentissage [ici] et d'un jeu de test [là]. Pré-traitez ces données (notez notamment que les termes sont spécifiés dans *GloVe* sous leur forme racinisée) pour obtenir un jeu de données prêt à être traité par un réseau de neurones récurrent.
2. Implémentez un réseau de neurones avec une couche récurrente (256 neurones) suivie de trois couches *fully connected* (128 et 64 neurones pour les deux premières, la troisième étant la couche de classification finale). Apprenez ce modèle en prenant bien soin d'éviter le sur-apprentissage et évaluez ses performances sur le jeu de test.