

*Master 2 IAAA*  
Cours de Deep Learning  
TD 6 - 2019-2020

T. Artières and QARMA



October 2019

I. Nombre de paramètres dans les modèles récurrents

On considère des modèles récurrents *SimpleRNN*, *LSTM*, et *GRU* de *keras* construits pour des données de dimension d'entrée de séquences de 12 vecteurs de dimension 72, et pour la classification des séquences en 8 classes.

- (1) On vous donne l'architecture suivante d'un réseau récurrent standard (*SimpleRNN* en *keras*). Dessinez le modèle en indiquant des noms pour les matrices de poids, les représentations dans les couches cachées, etc.

Layer (type)	Output Shape	Param #
input_41 (InputLayer)	(None, 12, 72)	0
simple_rnn_23 (SimpleRNN)	(None, 12, 100)	17300
simple_rnn_24 (SimpleRNN)	(None, 100)	20100
dense_28 (Dense)	(None, 8)	808

=====  
Total params: 38,208  
Trainable params: 38,208  
Non-trainable params: 0

- (2) Expliquez le nombre de paramètres du modèle *SimpleRNN* de la figure précédente.
- (3) Expliquez le nombre de paramètres du modèle *LSTM* suivant:

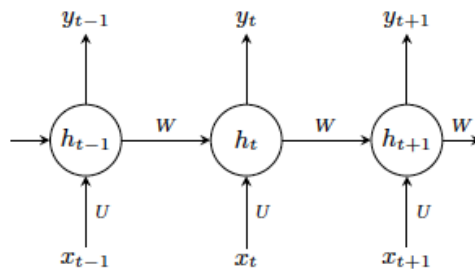
Layer (type)	Output Shape	Param #
input_42 (InputLayer)	(None, 12, 72)	0
lstm_21 (LSTM)	(None, 12, 100)	69200
lstm_22 (LSTM)	(None, 100)	80400
dense_29 (Dense)	(None, 8)	808
Total params: 150,408		
Trainable params: 150,408		
Non-trainable params: 0		

(4) Expliquez le nombre de paramètres du modèle *GRU* suivant:

Layer (type)	Output Shape	Param #
input_40 (InputLayer)	(None, 12, 72)	0
gru_9 (GRU)	(None, 12, 100)	51900
gru_10 (GRU)	(None, 100)	60300
dense_27 (Dense)	(None, 8)	808
Total params: 113,008		
Trainable params: 113,008		
Non-trainable params: 0		

## II. Propagation du gradient dans les RNN standards

On considère le modèle *RNN* suivant. L'état  $h_t$  est calculé suivant :  $h_t = \sigma(W h_{t-1} + U x_t)$  avec  $\sigma(z) = \frac{1}{1+e^{-z}}$ .



Soit  $L$  la fonction objectif définie comme la somme des  $L_t$  à chaque pas de temps:  $L = \sum_{t=1}^T L_t$ , où  $L_t$  est un terme de coût réel dépendant de  $y_t$  et donc de  $h_t$  (voir figure).

- (1) Soit  $u = \sigma(Av)$  avec  $u \in \mathbb{R}^n$ ,  $v \in \mathbb{R}^d$ ,  $W \in \mathbb{R}^{n \times d}$ . Montrez que le Jacobien  $\frac{\partial u}{\partial v}$  est égal à  $\text{Diag}(\sigma')W \in \mathbb{R}^{n \times d}$  où vous explicitez  $\text{Diag}(\sigma')$ .

(2) Montrer que  $\frac{\partial L}{\partial W}$  peut s'exprimer comme :  $\frac{\partial L}{\partial W} = \sum_{t=0}^T \sum_{k=1}^t \frac{\partial L}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial W}$ .

### III. Evanouissement / explosion du gradient dans les RNNs.

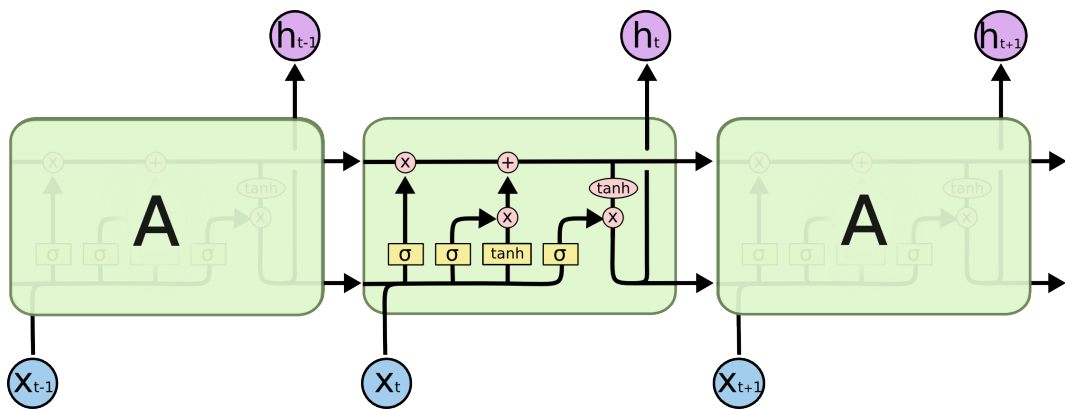
On repart de la seconde question de l'exercice précédent pour étudier sous quelles conditions la norme du gradient peut exploser ou s'évanouir si le nombre de couches augmente.

- (1) Ré-écrire  $\frac{\partial L}{\partial W}$  extensivement dans le cas où  $T = 3$ . Il devrait apparaître la matrice  $Diag(\sigma')W$  élevée à la puissance.
- (2) Faire apparaître la matrice  $Diag(\sigma')W$  élevée à la puissance dans le cas général de la seconde question de l'exercice précédent.
- (3) On rappelle que toute matrice carrée diagonalisable  $M$  peut s'exprimer, par décomposition en vecteurs propres, sous la forme  $M = Q\Lambda Q^{-1}$  où  $Q$  est une matrice dont la  $i^{ime}$  colonne est le  $i^{ime}$  vecteur propre de  $M$  et  $\Lambda$  est la matrice dans laquelle les valeurs propres correspondantes occupent la diagonale. Montrer que  $M^n = Q\Lambda^n Q^{-1}$ .
- (4) Qu'en déduisez vous sur le calcul du gradient dans un RNN si la longueur de la séquence d'entrée  $T$  est grande ? Que se passe-t-il si les valeurs propres sont (en valeur absolue) égales à 1, inférieures à 1, supérieures à 1 ?

### IV. LSTMs

On rappelle ci-dessous les équations régissant les cellules LSTM :

$$\begin{aligned}
 f_t &= \sigma(W_f h_{t-1} + U_f x_t) \\
 i_t &= \sigma(W_i h_{t-1} + U_i x_t) \\
 o_t &= \sigma(W_o h_{t-1} + U_o x_t) \\
 \tilde{c}_t &= \tanh(W_c h_{t-1} + U_c x_t) \\
 c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\
 c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\
 h_t &= o_t \odot \tanh(c_t)
 \end{aligned}$$



- (1) Que représentent:
  - La forget gate:  $f_t$ ?

- L'input gate:  $i_t$  ?
  - L'output:  $o_t$  ?
- (2) Pour calculer le gradient du loss  $L$  en sortie du réseau par rapport aux paramètres du réseau ( $W_c, W_i, W_o, \dots$ ) il faut pouvoir calculer, comme dans le cas du RNN simple le gradient par rapport à l'état  $c_t$ , qui dépend des valeurs de l'état aux instant précédents. Montrer que pour une des matrices de poids  $W$ , en négligeant certains termes on obtient:

$$\frac{\partial L}{\partial W} \approx \sum_{t=0}^T \sum_{k=1}^t \frac{\partial L}{\partial c_t} \frac{\partial c_t}{\partial c_k} \frac{\partial c_k}{\partial W}$$

- (3) Supposons que  $\forall t, f_t = 1$  et  $i_t = 0$ . Que vaut le gradient précédent dans ce cas ?
- (4) Exprimez dans le cas général  $\frac{\partial c_t}{\partial c_{t-1}}$ .