

Réseaux récurrents



Compétences travaillées : propagation avant dans un RNN, rétropropagation à travers le temps (BPTT), gradient vanishing, mécanisme de la cellule LSTM, comptage de paramètres (RNN, GRU, LSTM).

Version web

Ex 1

Propagation forward dans un RNN simple

pK4m

★ ★

On considère un RNN simple à un seul état caché, défini par les équations :

$$h_t = \tanh(W_h h_{t-1} + W_x x_t + b_h), \quad y_t = W_y h_t + b_y$$

avec les dimensions suivantes : entrée $x_t \in \mathbb{R}$, état caché $h_t \in \mathbb{R}^2$, sortie $y_t \in \mathbb{R}$.

Les paramètres du réseau sont :

$$W_x = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \quad W_h = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}, \quad b_h = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad W_y = (1 \quad 1), \quad b_y = 0$$

L'état caché initial est $h_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$. On traite la séquence d'entrée $x_1 = 1, x_2 = 0, x_3 = -1$.

1. Calculer h_1, h_2, h_3 en appliquant la formule de récurrence pas à pas. On rappelle que $\tanh$ s'applique composante par composante.
2. En déduire les sorties y_1, y_2, y_3 .
3. Décrire qualitativement ce que fait ce réseau : comment l'état caché h_t évolue-t-il lorsque l'entrée change de signe ?
4. (*Bonus*) Que se passerait-il si on remplaçait $\tanh$ par la fonction identité ? Réécrire alors h_t comme une combinaison linéaire des entrées passées $x_1, \dots, x_t$.

1

Développer la récurrence $h_t = W_h h_{t-1} + W_x x_t$ et itérer.

Ex 2

Rétropropagation à travers le temps (BPTT)

Tz9w

★ ★ ★

On considère un RNN simple scalaire ($h_t \in \mathbb{R}$) défini par :

$$h_t = \tanh(w_h h_{t-1} + w_x x_t), \quad y_t = w_y h_t$$

On dispose de la séquence $x_1 = 1, x_2 = 0, x_3 = -1$ et des cibles $\hat{y}_1 = 0, \hat{y}_2 = 0, \hat{y}_3 = 1$.

La fonction de perte est :

$$\mathcal{L} = \frac{1}{3} \sum_{t=1}^3 (y_t - \hat{y}_t)^2$$

Les paramètres sont $w_x = 1, w_h = 0.5, w_y = 1$, et $h_0 = 0$.

1. (*Forward pass*) Calculer h_1, h_2, h_3 , puis y_1, y_2, y_3 , et enfin la perte $\mathcal{L}$.
2. Calculer $\frac{\partial \mathcal{L}}{\partial y_t}$ pour chaque t .
3. On pose $z_t = w_h h_{t-1} + w_x x_t$ (l'entrée du neurone avant activation) et $\delta_t = \frac{\partial \mathcal{L}}{\partial z_t}$. Montrer que :

$$\delta_t = \left(\frac{\partial \mathcal{L}}{\partial y_t} \cdot w_y + \delta_{t+1} \cdot w_h \right) (1 - h_t^2)$$

avec la convention $\delta_4 = 0$. Calculer numériquement $\delta_3, \delta_2, \delta_1$ en remontant le temps.

1

Appliquer la règle de dérivation en chaîne, en considérant que la perte dépend de z_t à la fois directement (via y_t) et indirectement (via l'état suivant z_{t+1}).

4. En déduire les gradients par rapport aux poids :

$$\frac{\partial \mathcal{L}}{\partial w_y} = \sum_{t=1}^3 \frac{\partial \mathcal{L}}{\partial y_t} h_t, \quad \frac{\partial \mathcal{L}}{\partial w_h} = \sum_{t=1}^3 \delta_t h_{t-1}, \quad \frac{\partial \mathcal{L}}{\partial w_x} = \sum_{t=1}^3 \delta_t x_t$$

Calculer numériquement ces trois gradients.

5. Le calcul de δ_t fait intervenir un produit récurrent en remontant le temps. Que se passe-t-il si la séquence est très longue ? Pourquoi le fait d'avoir $w_h < 1$ est-il un problème ?

Ex 3

Le problème du gradient vanishing

Hv2R

★ ★ ★

On considère un RNN simple scalaire sans entrée, pour isoler la dynamique de l'état caché :

$$h_t = \tanh(w_h h_{t-1})$$

On s'intéresse à la propagation du gradient sur T pas de temps. Par la règle de dérivation en chaîne :

$$\frac{\partial h_T}{\partial h_0} = \prod_{t=1}^T \frac{\partial h_t}{\partial h_{t-1}} = \prod_{t=1}^T w_h \cdot (1 - h_t^2)$$

1. Justifier la formule ci-dessus en appliquant la règle de dérivation en chaîne à $h_T = f(h_{T-1}) = f(f(\dots f(h_0) \dots))$.
2. Montrer que pour tout $h \in \mathbb{R}$, on a $0 < 1 - \tanh(h)^2 \leq 1$, avec égalité uniquement en $h = 0$. Que peut-on en conclure sur chaque facteur $w_h \cdot (1 - h_t^2)$ lorsque $|w_h| \leq 1$?
3. On suppose $h_t \approx 0.9$ pour tout t , et $w_h = 0.8$. Calculer numériquement $\left| \frac{\partial h_T}{\partial h_0} \right|$ pour $T = 5, T = 10, T = 20$. Commenter.
4. Montrer que le gradient ne s'évanouit pas si et seulement si $|w_h| \cdot (1 - h_t^2) \geq 1$ pour tout t . Quelle condition sur w_h cela impose-t-il lorsque $h_t \approx 0$? Quel nouveau problème apparaît alors ?
5. On cherche à minimiser une perte $\mathcal{L}$ dépendant de h_T . Exprimer $\frac{\partial \mathcal{L}}{\partial w_h} \Big|_{t=1}$, la contribution

du tout premier pas de temps au gradient de w_h , en fonction de $\frac{\partial \mathcal{L}}{\partial h_T}$ et du produit ci-dessus. Pourquoi les dépendances à long terme sont-elles si difficiles à apprendre ?

Ex 4

La cellule LSTM à la main

Xb5N

★ ★

Une cellule LSTM maintient deux vecteurs d'état : la *cell state* $c_t \in \mathbb{R}$ et l'état caché $h_t \in \mathbb{R}$ (on reste scalaire pour les calculs). À chaque pas de temps, quatre quantités intermédiaires sont calculées à partir de x_t et h_{t-1} :

$$\begin{aligned} f_t &= \sigma(w_f x_t + u_f h_{t-1} + b_f) && \text{(porte d'oubli)} \\ i_t &= \sigma(w_i x_t + u_i h_{t-1} + b_i) && \text{(porte d'entrée)} \\ \tilde{c}_t &= \tanh(w_c x_t + u_c h_{t-1} + b_c) && \text{(candidat)} \\ o_t &= \sigma(w_o x_t + u_o h_{t-1} + b_o) && \text{(porte de sortie)} \end{aligned}$$

La mise à jour des états suit :

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t, \quad h_t = o_t \cdot \tanh(c_t)$$

où σ désigne la fonction sigmoïde.

Les paramètres sont tous les biais nuls et :

	w	u
Porte d'oubli f	0	0
Porte d'entrée i	2	0
Candidat $\tilde{c}$	2	0
Porte de sortie o	1	0

Les états initiaux sont $h_0 = 0$, $c_0 = 0$. On traite la séquence $x_1 = 1$, $x_2 = 1$, $x_3 = 0$, $x_4 = -1$. Pour gagner du temps, vous pouvez utiliser : $\sigma(0) = 0.5$, $\sigma(1) \approx 0.731$, $\sigma(-1) \approx 0.269$, $\sigma(2) \approx 0.88$, $\sigma(-2) \approx 0.12$, $\tanh(2) \approx 0.96$, $\tanh(0) = 0$.

- (Pas $t = 1$) Calculer $f_1, i_1, \tilde{c}_1, o_1$, puis c_1 et h_1 .
- (Pas $t = 2$) Calculer $f_2, i_2, \tilde{c}_2, o_2$, puis c_2 et h_2 . Que remarque-t-on sur c_2 par rapport à c_1 ?
- (Pas $t = 3$) L'entrée est $x_3 = 0$. Calculer c_3 et h_3 . Que fait la porte d'oubli ici ? Comparer c_3 et c_2 .
- (Pas $t = 4$) L'entrée est $x_4 = -1$. Calculer c_4 et h_4 . Commenter le rôle de la porte d'entrée.
- Dans un RNN simple, l'état caché h_t était directement écrasé à chaque pas. Expliquer en quoi le mécanisme de cell state c_t de la LSTM permet de mieux préserver l'information sur le long terme.

Ex 5

Compter les paramètres d'un RNN, LSTM et GRU

Lq8D

Le nombre de paramètres d'un réseau est un indicateur clé de sa capacité et de son coût d'entraînement. On note :

- d : dimension de l'entrée x_t ,
- h : dimension de l'état caché,
- p : dimension de la sortie y_t .

Chaque architecture possède une **couche de sortie linéaire** $y_t = W_y h_t + b_y$.

Rappel des architectures.

RNN simple :

$$h_t = \tanh(W_x x_t + W_h h_{t-1} + b_h)$$

LSTM — quatre portes, chacune de la forme $W x_t + U h_{t-1} + b$:

$$f_t, \quad i_t, \quad \tilde{c}_t, \quad o_t$$

GRU — trois portes :

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (\text{porte de réinitialisation})$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (\text{porte de mise à jour})$$

$$\tilde{h}_t = \tanh(W_h x_t + U_h (r_t \odot h_{t-1}) + b_h) \quad (\text{candidat})$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

1. Justifier que le nombre de paramètres du RNN simple (hors couche de sortie) est $N_{\text{RNN}} = h(d + h + 1)$. Détailler la contribution de chaque matrice et biais.
2. Montrer que le nombre de paramètres de la LSTM (hors couche de sortie) est $N_{\text{LSTM}} = 4h(d + h + 1)$. Pourquoi ce facteur 4 ?
3. Montrer que le nombre de paramètres du GRU (hors couche de sortie) est $N_{\text{GRU}} = 3h(d + h + 1)$.
4. Ajouter la contribution de $W_y \in \mathbb{R}^{p \times h}$ et $b_y \in \mathbb{R}^p$. Écrire le nombre total de paramètres N_{total} pour chacune des trois architectures.
5. Application numérique : $d = 10$, $h = 64$, $p = 5$. Compléter le tableau suivant.

Architecture	Paramètres (hors sortie)	Paramètres (avec sortie)
RNN simple		
GRU		
LSTM		

6. La LSTM a davantage de paramètres que le GRU, mais est-elle toujours plus performante ? Dans quel cas préférerait-on le GRU ?