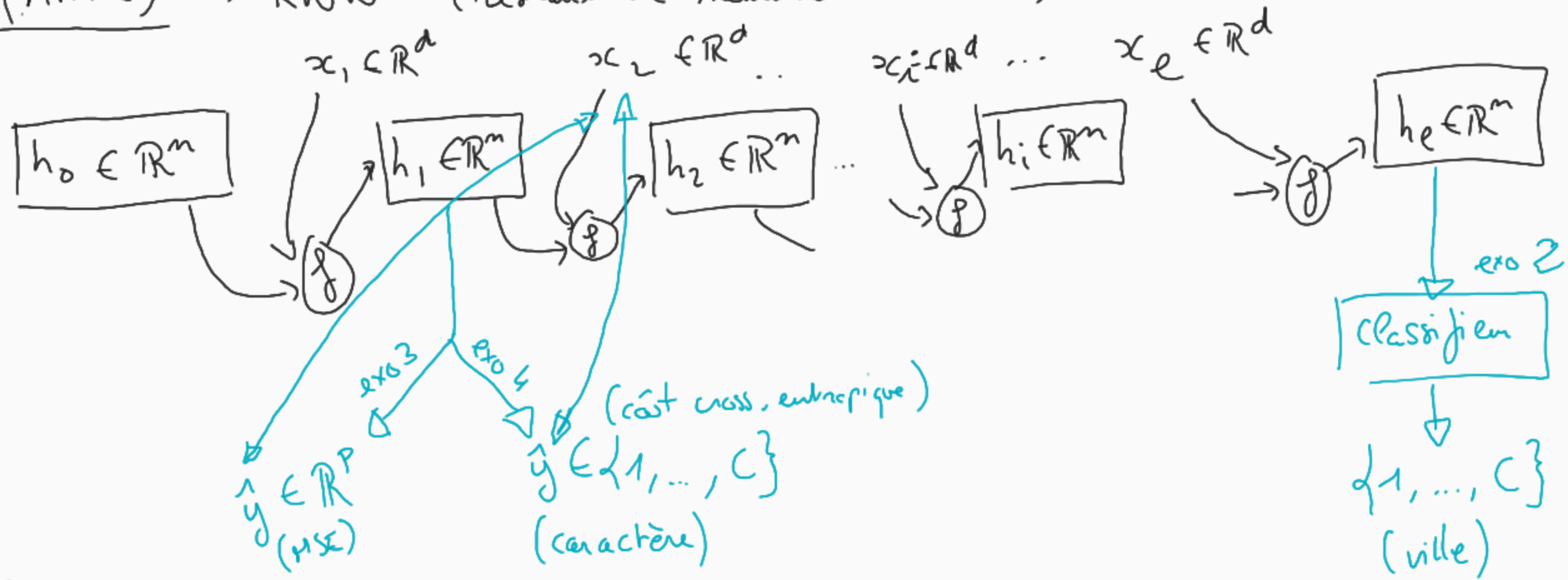
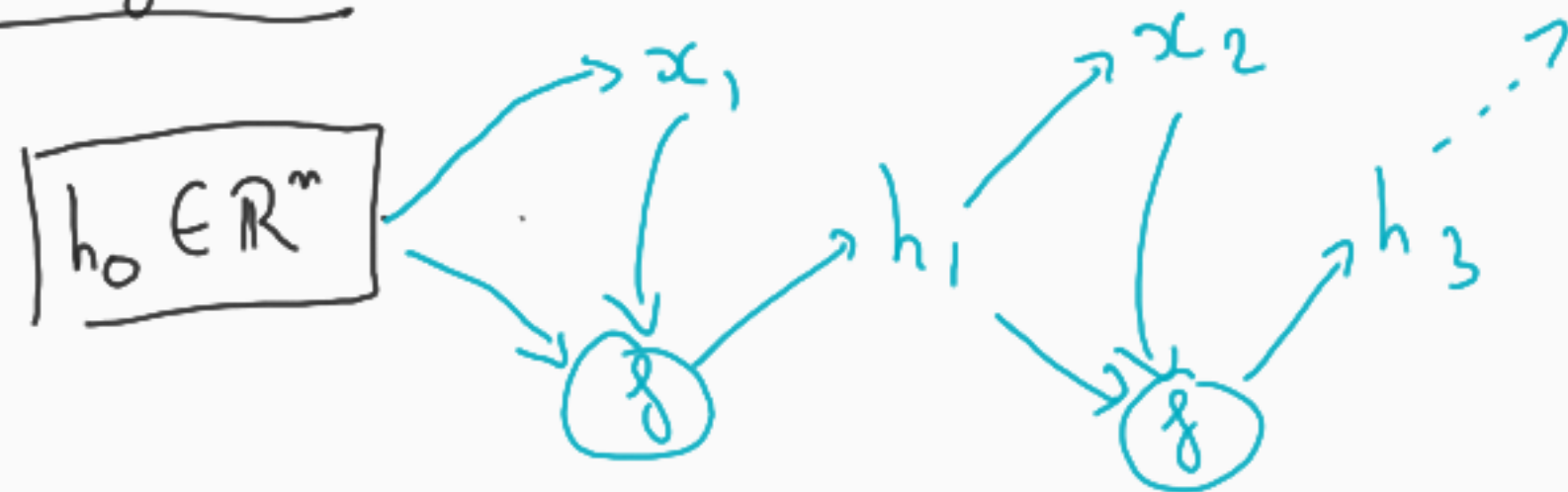


TME 4 (ANAL) → RNN (réseaux de neurones récurrents)



En inférence



En pratique (Torch)

Batch b
Langueur l
Dimension d

Tenseur

$\mathbb{R}^{l \times b \times d}$

pour optimiser au mieux
l'accès mémoire

Calculer les h_1, \dots, h_l d'une seule traite = forward

$\mathbb{R}^{b \times m}$
 $\underbrace{\hspace{1cm}}_{h_0}$ / $\mathbb{R}^{l \times b \times d}$
 $\underbrace{\hspace{1cm}}_{\text{observations}}$

$\mathbb{R}^{l \times b \times m}$
 h_1, \dots, h_l

classes $\hat{y}_1, \dots, \hat{y}_l \in \{1, \dots, \beta\}$
obs. $\hat{y}_1, \dots, \hat{y}_l \in \mathbb{R}^d$

exo 1

pour apprendre, comparer
les prédictions aux observations



Il faut prédire
 y_1 à partir
de h_0 , puis
 y_2 à partir
de h_1, \dots

$$x \in \mathbb{R}^{l \times b \times d}$$

$$h \in \mathbb{R}^{b \times n} \quad \leftarrow \text{taille de l'espace latent (à choisir)}$$

$$\bullet t \leftarrow []$$

$$\bullet \text{ Boucle } i = 1, \dots, l$$

$$\left[\begin{array}{l} h_{\text{new}} \leftarrow \text{one_step}(x[i], h) \\ t \leftarrow t \oplus h_{\text{new}} \\ h \leftarrow h_{\text{new}} \end{array} \right.$$

• Utilisez

torch.cat / torch.stack

$$\rightarrow \mathbb{R}^{l \times b \times n}$$

$$= [h_1, \dots, h_l]$$

$t = \text{liste de tensors de taille } \mathbb{R}^{b \times n} \text{ (l'éléments)}$

$$\text{torch.cat} \quad (2 \times 3), (2 \times 3) \rightarrow (4 \times 3)$$

$$\text{torch.stack} \quad (2 \times 3), (2 \times 3) \rightarrow (2 \times 2 \times 3)$$

(Page 1)

$$\underbrace{h_{t+1}}_{\in \mathbb{R}^n} = f(x_t, h_t) = \tanh \left(\underbrace{W_i x_t}_{\substack{\uparrow \\ \mathbb{R}^d \rightarrow \mathbb{R}^n}} + \underbrace{W_h h_t + b_h}_{\substack{\uparrow \\ \mathbb{R}^n \rightarrow \mathbb{R}^n}} \right) \in \mathbb{R}^n$$

W_i, W_h, b_h



partagé par tous
les pas de temps