

## Instructions

Ecrire l'examen avec un traitement de texte, e.g. avec Overleaf.

Vous avez une semaine pour le faire, i.e. rendu le 9/12/2020

M'envoyer ensuite le pdf par courrier, [patrick.gallinari@sorbonne-universite.fr](mailto:patrick.gallinari@sorbonne-universite.fr)

## Modèles de langue neuronaux

Le sujet porte sur des modèles de langue neuronaux. Il s'agit d'apprendre des représentations des mots en contexte à partir de corpus de texte. On s'intéresse à des modèles qui « lisent » un mot et doivent prédire le contexte du mot dans la phrase. Le contexte peut être défini de différentes façons : le mot suivant dans la phrase, les mots qui entourent le mot courant dans la phrase, etc.

On dispose d'un corpus  $C$  constitué de phrases, d'un dictionnaire de  $n$  mots  $Dict = \{w_1, \dots, w_n\}$ . On veut construire un modèle neuronal qui étant donné un mot  $w_I$  dans une phrase, doit prédire le mot suivant dans la phrase, noté  $w_O$ . On considère le modèle neuronal de la figure 1.

Le mot  $w_i$  sera codé par un vecteur « 1 parmi  $n$  » noté  $x \in \{0,1\}^n$  avec un 1 en position  $i$  et 0 partout ailleurs. Le réseau comprend  $n$  entrées et  $n$  sorties correspondant chacune à un mot du dictionnaire.  $U$  et  $V$  sont des matrices de poids de dimensions respectives  $d \cdot n$  et  $n \cdot d$ . Le vecteur  $u_i \in R^d$  est la  $i^{eme}$  colonne de  $U$  et  $v_j \in R^d$  est la  $j^{eme}$  ligne de  $V$ . Pour une entrée  $w_I$  codée par  $x$ , on calcule la sortie du réseau de la façon suivante : on calcule d'abord la représentation cachée  $Ux = u_I$ , puis la sortie correspondant au mot  $w_k$ ,  $y_k = \frac{\exp(v_k \cdot u_I)}{\sum_{j=1}^n \exp(v_j \cdot u_I)} = p(w_k | w_I)$  pour tous les mots  $w_k \in Dict$ . La notation  $v \cdot u$  désigne le produit scalaire des deux vecteurs  $v$  et  $u$ . On reconnaît un « softmax » calculé sur les cellules de sortie.

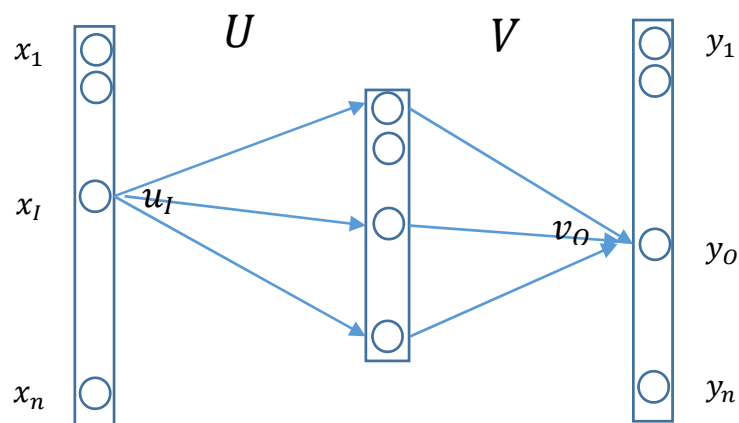


Figure 1

Les paramètres du modèle sont les  $u_i, i = 1 \dots n$  et  $v_j, j = 1 \dots n$ .  $u_i$  et  $v_i$  sont deux représentations différentes du mot  $w_i$ . On dira que  $u_i$  est la représentation d'entrée et  $v_i$  est la représentation de sortie.  $y_k$  est un estimateur de la probabilité a posteriori ( $w_k | w_I$ ).

On note  $D$  l'ensemble des  $(w_I, w_O)$  qui forment un couple mot d'entrée – mot de contexte. Dans notre exemple,  $D$  sera constitué de tous les mots rencontrés dans les phrases et de leur suivant dans la phrase. Le critère d'apprentissage est la vraisemblance conditionnelle des observations  $(w_I, w_O)$ .

La fonction de coût est la log vraisemblance  $L = \sum_{(w_I, w_O) \in D} \log p(w_O | w_I)$ . On note  $e(w_I, w_O) = -\log p(w_O | w_I)$ .

1. Quel est l'effet de la maximisation de la log vraisemblance  $L$  sur les vecteurs  $u_I, v_O$  ?
2. Si deux mots  $w_I, w_J$  apparaissent souvent avec les mêmes mots de contexte  $w_O$ , que pouvez-vous dire sur les représentations apprises  $u_I, u_J$  de ces mots ?
3. Dérivation d'un algorithme d'apprentissage

On va utiliser un algorithme de gradient stochastique : à chaque itération, on tire un exemple  $(w_I, w_O)$  et on utilise le gradient stochastique pour maximiser  $e$ .

3.1 Donner l'expression de  $e(w_I, w_O) = -\log p(w_O | w_I)$  en fonction des paramètres  $u$  et  $v$ .

3.2 On note  $e = e(w_I, w_O)$ ,  $a_i = v_i \cdot u_I$  et  $b_i = \sum_{j=1}^n u_{ij} x_{Ij} = u_{iI}$

En utilisant la règle de dérivation  $\frac{\partial e}{\partial v_{ij}} = \frac{\partial e}{\partial a_i} \cdot \frac{\partial a_i}{\partial v_{ij}}$  donner l'expression de  $\frac{\partial e}{\partial v_{ij}}$

En utilisant la règle de dérivation  $\frac{\partial e}{\partial u_{ij}} = \frac{\partial e}{\partial b_i} \cdot \frac{\partial b_i}{\partial u_{ij}}$  et  $\frac{\partial e}{\partial b_i} = \sum_{k=1}^n \frac{\partial e}{\partial a_k} \frac{\partial a_k}{\partial b_i}$  donner l'expression de  $\frac{\partial e}{\partial u_{ij}}$

3.3 Proposer un algorithme de gradient stochastique pour mettre à jour les poids du réseau.

4. Analyse de la complexité.

Lors de l'apprentissage, on échantillonne les couples  $(w_I, w_O)$ , et pour chaque couple on calcule  $p(w_O | w_I)$ . Quelle est la complexité de ce calcul ?

Quelle est la complexité de la mise à jour des poids pour un couple d'exemple ?

Voyez-vous des alternatives pour réduire cette complexité ?