Examen Apprentissage Statistique

Master Informatique UPMC – spécialité DAC

26-01-2016

Notes de cours autorisées

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 ont le fonctionnement suivant : ils « 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.

A. Modèle neuronal

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.

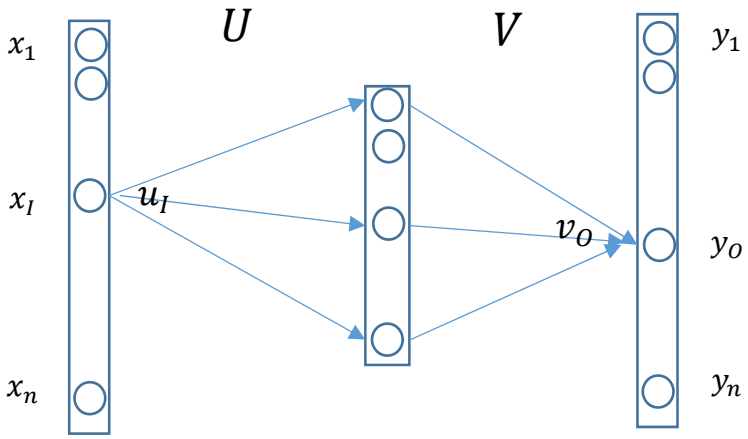

Figure 1

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 $d.n$ et $n.d$. 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, $y_k = \frac{\exp(v_k.u_I)}{\sum_{j=1}^{n} \exp(v_j.u_I)} = p(w_k|w_I)$ pour tous les mots $w_k \in Dict$. La notation $v.u$ désigne le produit scalaire des deux vecteurs v et u. On reconnait un « softmax » calculé sur les cellules de sortie.

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 $\mathrm{e}(\mathrm{w_I}, \mathrm{w_O}) = \log p(w_O|w_I)$.

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

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

En utilisant la règle de dérivation $\frac{\partial e}{\partial u_{ij}} = \frac{\partial e}{\partial b_i}.\frac{\partial b_i}{\partial u_{ij}}$ 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 ?

B. Echantillonnage négatif

On introduit une famille de méthodes pour réduire cette complexité. On formalise le problème comme un problème de classification. On considère l'ensemble des couples $(w_I, w_O) \in D,$ et on génère suivant une certaine distribution un ensemble de couples (w_I, w) qui ne sont pas D(des exemples négatifs), on note D' ce deuxième ensemble. On note $p(c = 1|w_I, w)$ la probabilité que (w_I, w) soit dans D et $p(c = 0|w_I, w) = 1 - p(c = 1|w_I, w)$ la probabilité qu'il ne soit pas dans D.

On modélise $p(c = 1|w_I, w)$ par $p(c = 1|w_I, w) = \sigma(v.u_I)$. où v et u_I sont les représentations des mots w et w_I, et $\sigma(x) = \frac{1}{1+\exp(-x)}$.

On remplace le problème d'apprentissage initial par la maximisation de la vraisemblance :

$$l = \prod_{(w_I,w)\in D} P(c = 1|w_I, w) \prod_{(w_I,w)\in D'} P(c = 0|w_I, w)$$

1. Montrer $L = \log l = \sum_{(w_I,w_O)\in D} \log \sigma(v_O.u_I) + \sum_{(w_I,w)\in D'} \log \sigma(-v_w.u_I)$

2. Il existe différentes façons d'échantillonner les exemples dans D', on considère par la suite une méthode dite d'échantillonnage négatif qui consiste pour chaque exemple $(w_I, w_O) \in D$ à générer k exemples (w_I, w) où w est tiré de la distribution des mots du corpus $P(w)$. On considère alors la fonction suivante :

$$e(w_I, w_O) = \log \sigma(v_o.u_I) + kE_{w\sim P(w)}(\log \sigma(-v_w.u_I)) \quad (1)$$

Où v_w est le vecteur de sortie pour le mot w, $E_{w\sim P(w)}$ est l'espérance calculée sur cette distribution. En pratique, on ne calcule pas l'espérance, mais à chaque fois que le couple (w_I, w_O) est sélectionné, on échantillonne k mots $w_1, \dots, w_k$ suivant cette distribution et l'on optimise

$$e(w_I, w_O) = \log \sigma(v_o.u_I) + \sum_{i=1}^{k} \log \sigma(-v_{w_i}.u_I))$$

Quelle est la complexité d'une itération de cet algorithme pour une itération d'apprentissage (un exemple) ?

C. Echantillonnage négatif et factorisation matricielle

On considère la matrice $M = VU$ où U et V sont définies comme plus haut. Le produit VU peut être vu comme une factorisation matricielle de la matrice M (taille $n.n$) par deux matrices de taille inférieure (resp. $(n.d)$ et $(d.n)$). On essaie de trouver une réponse à la question : quelle matrice M, U et V factorisent elles ?

1. Information mutuelle ponctuelle

Un mesure d'association entre termes d'un corpus est l'information mutuelle ponctuelle entre 2 termes w et w' : $PMI(w, w') = \log \frac{p(w,w\prime)}{p(w)p(w\prime)}$, cette valeur est maximale si $p(w|w')$ ou $p(w'|w) = 1$, i.e. les termes sont parfaitement associés, elle est nulle si ils sont indépendants.

On considère le corpus D des couples (w_I, w_O) que l'on veut associer. On note $f(w)$ et $f(w, w')$ le nombre d'apparitions de w et (w, w') dans D.

Donner l'expression de $PMI(w, w')$ en fonction de ces fréquences.

2. En reprenant l'expression (1), le coût global optimisé est

$$L = \sum_{w_I \in Dict} \sum_{w_O \in Dict} f(w_I, w_O)(\log \sigma(v_O.u_I) + kE_{w\sim P(w)}[\log \sigma(-v_w.u_I)])$$

2.1 Montrer

$$L = \sum_{w_I \in Dict} \sum_{w_O \in Dict} f(w_I, w_O) \log \sigma(v_O.u_I) + \sum_{W_I \in Dict} f(w_I)(kE_{w\sim P(w)}[\log \sigma(-v_w.u_I)])$$

2.2 On estime l'espérance par $E_{w\sim P(w)}[\log \sigma(-v_w.u_I)] = \sum_{w \in Dict} \frac{f(w)}{|D|} \log \sigma(-v_w.u_I)$

On cherche les termes de la matrice M qui optimisent L. Chacun de ces termes correspond à un produit $v_O.u_I$ i.e. à un couple (w_I, w_O). On va isoler dans L l'expression qui dépend de ce produit $v_O.u_I$.

Montrer que cette expression s'écrit

$$L(w_I, w_O) = f(w_I, w_O) \log \sigma(v_O.u_I) + kf(w_I).\frac{f(w_O)}{|D|} \log \sigma(-v_O.u_I)$$

2.3 Notons $a = v_O.u_I$, calculer $\frac{\partial L(w_I, w_O)}{\partial a}$

On se rappellera que $\sigma'(x) = \sigma(x)(1 - \sigma(x))$ et $\sigma(-x) = 1 - \sigma(x)$

2.4 Résoudre $\frac{\partial L(w_I, w_O)}{\partial a} = 0$, on posera $a' = \exp(a)$, on exprimera $\frac{\partial L(w_I, w_O)}{\partial a} = 0$ comme une équation du 2nd degré en a' que l'on résoudra et on résoudra d'abord l'équation en a'.

2.5 Montrer $v_O.u_I = \log(\frac{f(w_I, w_O).|D|}{f(w_I)f(w_O)}) - \log k$

2.6 Quelle relation constatez-vous avec l'information mutuelle ponctuelle ? Quelle interprétation peut-on donner à l'algorithme proposé ?