Deep learning for natura language processing

Guillaume PETIT

1 Multilingual word embeddings

Soit X et Y deux matrices dans $\mathbb{R}^{dxm}$ avec m la taille du vocabulaire. On veut résoudre le problème d'optimisation :

$$\min_w ||WX - Y||_F^2$$
$$\text{sc } W^T W = I_d$$

Or, $||WX - Y||_F^2 = ||WX||_F^2 + ||Y||_F^2 - 2 < WX, Y >_F$. Puisque W est orthogonale,

$$||WX||_F^2 = Tr\left(X^T W^T W X\right) = Tr\left(X^T X\right) = ||X||_F^2$$

Pour résoudre notre problème, on allons donc maximiser $< WX, Y >_F$. Soient U, V orthogonales et $\Sigma \geqslant 0$ et diagonale tel que $YX^T = U\Sigma V^T$. Par la propriété de la trace, on a $< W, U\Sigma V^T >_F =< U^T WV, Y >_F$. La matrice $H = U^T WV$ est orthogonale par produit de matrices orthogonales et par l'inégalité de Cauchy-Schwarz,

$$< WX, Y >_F =< W, YX^T >_F =< H, \Sigma > \quad \leqslant \quad Tr\left(\Sigma\right)$$

avec égalité ssi $H = I_d$, c'est-à-dire $W = UV^T$.

2 Sentence classification with BoW

On tune le paramètre de régularisation de notre régression logistique C, avec $C \in \{0.01, 0.05, 0.1, 0.5, 1, 2\}$, avec et sans idf. On essaie nos propres modèles, comme une random forest et un réseau de neurones. On obtient les résultats suivants :

modèles	paramètres	score	score idf
RL	0.01	Train : 0.46 % dev : 0.41%	Train : 0.45% dev : 0.40 %
MLP	-	Train : 0.58 % dev : 0.387%	Train : 0.63% dev : 0.384 %
RF	-	Train : 0.88 % dev : 0.22%	Train : 0.89% dev : 0.24 %

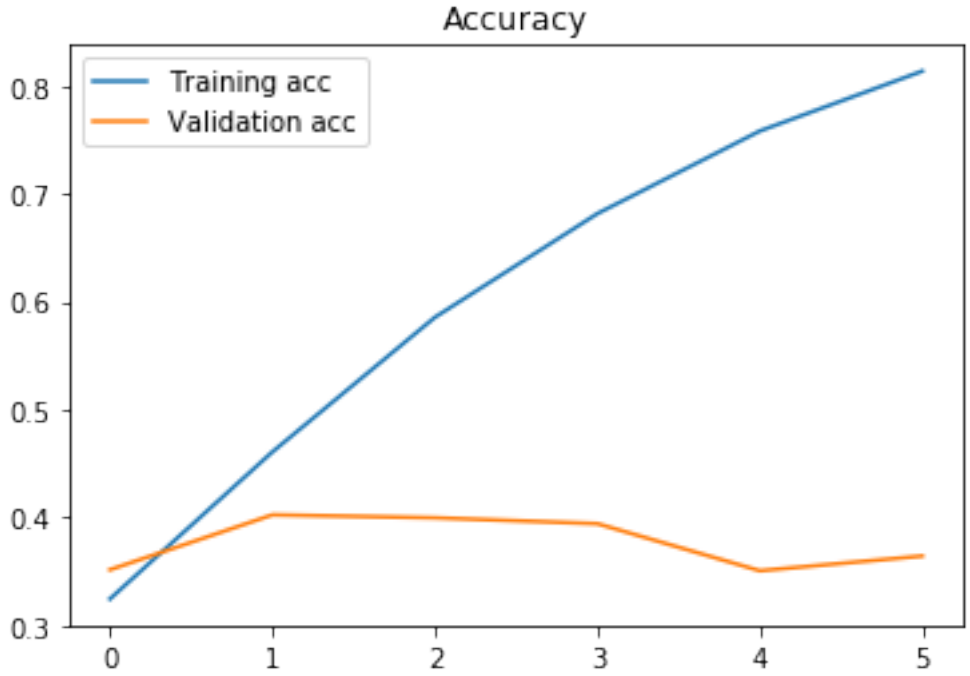
3 Deep Learning models for classification

question 1) J'ai utilise la perte "cross-entropy" :

$$\sum_{i=0}^{4} \mathbf{1}_{y=i} log(p_i)$$

avec k la classe de y et p_i la probabilité prédite par le modèle si l'observation y appartient à la classe i

question 2) Après avoir fait varier quelques hyper-paramètres, comme la taille de l'embedding ou du nombre d'unités cachées du LSTM, on obtient les résultats suivants :

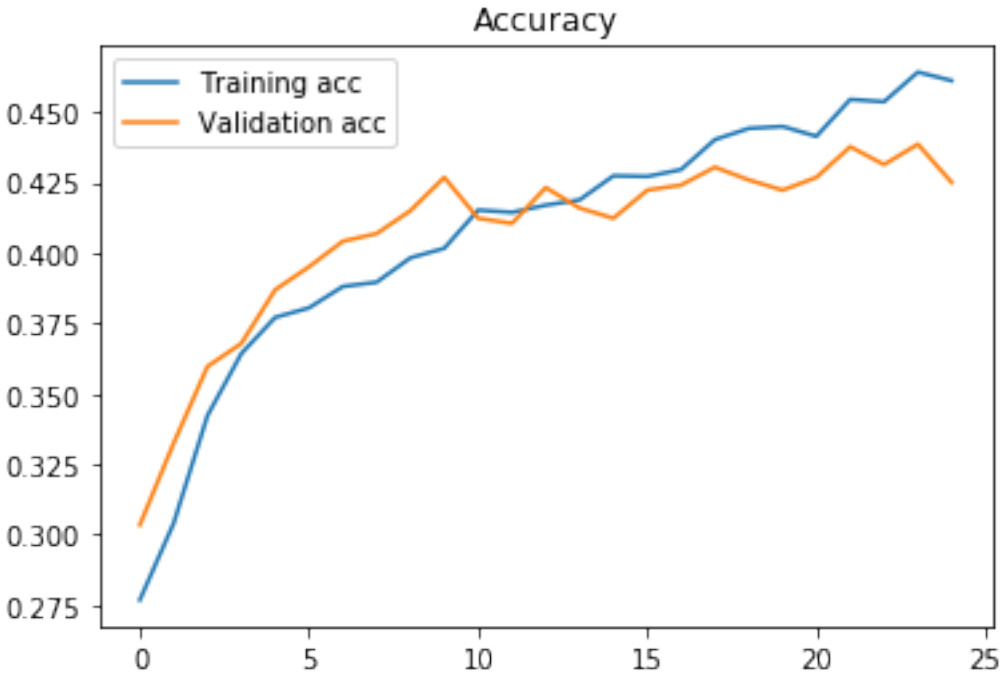


accuracy du modèle

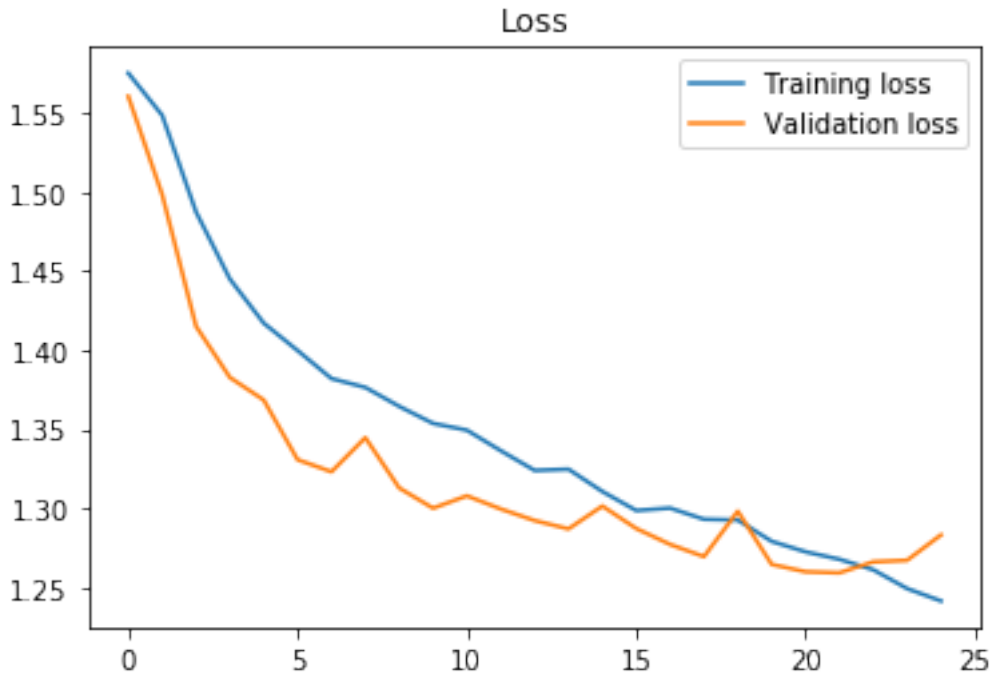
perte du modèle du modèle

question 3) On remarque que le modèle précédent à tendance à overfiter trop rapidement. Cela est sûrement dû à la taille de notre vocabulaire, et donc de notre matrice d'embedding, comparé à la taille de notre ensemble

de training. Ainsi, je vais donc réduire la taille de notre matrice d'embedding. On remarque qu'on a bien un modèle qui n'overfit pas. J'ai essayé des modèles avec des Conv1D, des LSTM bi-directionnels, mais cela n'amélioraient pas les scores de manières significatives.



accuracy du modèle



perte du modèle du modèle