RETHINKING ATTENTION WITH PERFORMERS

Jean Soler, Olivier Serris, Nicolas Castanet
Sorbonne Université - M2 DAC

1 - Le problème de l'Attention

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d}}\right)V$$

Complexité de calcul matricielle AB avec $A \in \mathbb{R}^{nm}$, $B \in \mathbb{R}^{mp}$: $\mathcal{O}(nmp)$

Dans l'attention, on a les matrices **Querys, Keys et Values** :
$Q \in \mathbb{R}^{Ld}$, $K^T \in \mathbb{R}^{dL}$ et $V \in \mathbb{R}^{Ld}$

$\begin{cases} L : \text{taille des séquences (possiblement très grand !)} \\ d : \text{dimension d'embedding} \end{cases}$

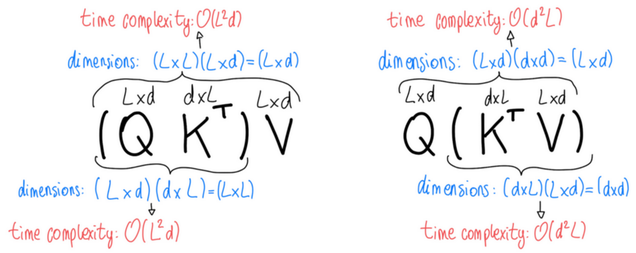


Fig. 1: complexité de calcul en fonction de l'ordre de la multiplication matricielle

$\implies$ **Le softmax force la complexité de calcul à être quadratique en L :** $\mathcal{O}(L^2 d)$

2 - Approximation du Softmax

Objectif : trouver $Q^{'}$ et $K^{'}$ tel que :

$$Q^{'}K^{'T} \approx Softmax\left(\frac{QK^T}{\sqrt{d}}\right)$$

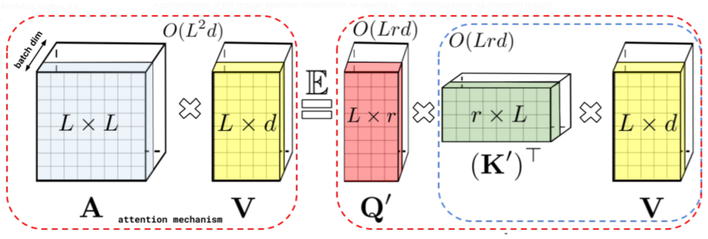
tel que

$$Attention(Q, K, V) \approx Q^{'}(K^{'T}V)$$

Réécriture du Softmax : $Softmax\left(\frac{QK^T}{\sqrt{d}}\right) = D^{-1}A$ avec,

$$\begin{cases} A = exp(QK^T/\sqrt{d}) \\ D = diag(A1_L) \end{cases}$$

$$\implies Attention(Q, K, V) = D^{-1}AV$$



3 - Fonctions noyaux

On veut trouver une fonction noyaux ϕ telle que :

$$K_{SM}(x, y) = \exp(xy^T) = \phi(x)^T\phi(y)$$

En utilisant l'approximation du noyaux gaussien grâce aux features de fourriers, les auteurs de [1] trouvent :

$$K_{SM}(x, y) = \mathbb{E}_{w\sim\mathcal{N}(0,I_d)}[\exp(-\frac{||x||^2}{2})\exp(-\frac{||y||^2}{2})\exp(w^Tx)\exp(w^Ty)]$$

$$\implies \phi(x) = \frac{1}{\sqrt{R}}\exp(-\frac{||x||^2}{2})(\exp(w_1^Tx), ..., exp(w_R^Tx))$$

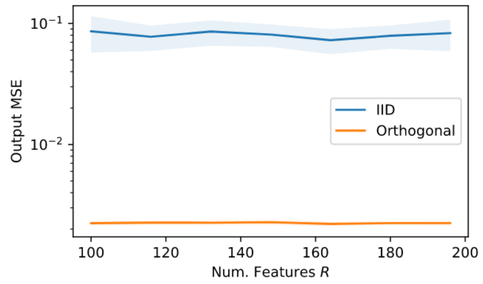
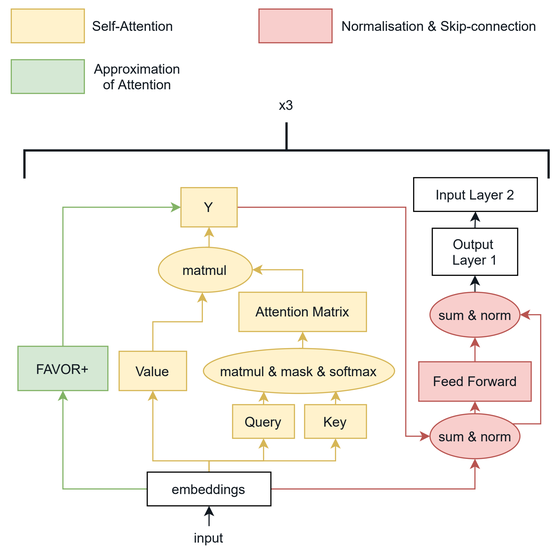


Fig. 3: MSE loss entre l'approximation et le softmax selon le nombre de features (iid vs orthogonales), embedding dim d = 100

4 - Architecture du modèle



5 - Temps execution

Dans l'expérience suivante on étudie le temps d'exécution d'une passe (forward + backward) à travers une couche d'attention avec une seule tête.

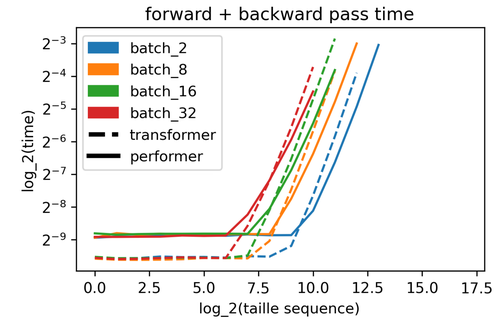


Fig. 5: num Features R = 100, embedding dim d = 50

Sur des séquences de grandes tailles le performer permet d'améliorer la vitesse de calcul et/ou d'augmenter la taille du batch.

6 - Accuracy sur IMBD

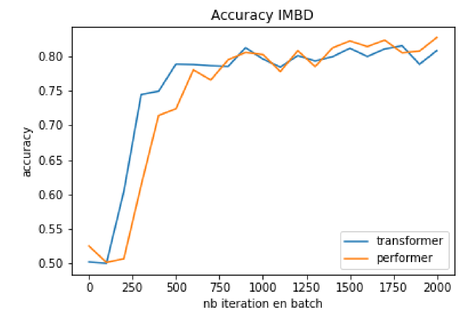


Fig. 6: Courbe d'accuracy sur IMBD. (orange = transformer, bleu = performer) , num Features R = 100, embedding dim d = 50

Les performances d'un transformer VS un performer ne sont pas différentiables sur une base simple comme IMDB.
Pour le même nombre d'epoch l'apprentissage est 50% plus rapide chez le performer.

References

[1] Krzysztof et al. *Rethinking Attention with Performers* - ICLR 2021
[2] Vaswani et al. *Attention Is All You Need* - 2017
[2] Yu et al. *Orthogonal Random Features* - 2016
[3] Rahimi et al. *Random Features for Large-scale Kernel Machines*