TD 2020 DAC - AMAL

Skip Gram avec embeddings gaussiens

1. En ½ page, le principe du modèle skip gram pour les embeddings de mots.

Au lieu de représenter les mots par des vecteurs, on va les représenter par une distribution Gaussienne : à chaque mot on associe une moyenne et une matrice de variance-covariance. Le modèle est plus riche et permet de représenter des incertitudes liées à la représentativité des données. On note w un terme du dictionnaire, et $c(w)$ un contexte de w. Par la suite, ce contexte sera réduit à 1 terme uniquement. Au mot w_i on associera la représentation $z_i \sim \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_i, \Sigma_i)$, $z_i \in R^n$. Le modèle skip-gram emploie le produit scalaire pour mesurer la similarité de deux représentations vectorielles. Le produit scalaire sera remplacé ici par une similarité entre deux distributions. On considère par la suite deux alternatives pour cela.

2. Similarité symétrique : noyau produit de probabilités.

Pour deux distributions f et g, le noyau, qui étend le produit scalaire vectoriel, est défini par $K(f,g) = \int_{z \in R^n} f(\boldsymbol{z}) g(\boldsymbol{z}) d\boldsymbol{z}$. On note pour simplifier $K\left(\mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_i, \Sigma_i), \mathcal{N}\left(\boldsymbol{z}; \boldsymbol{\mu}_j, \Sigma_j\right)\right) = K(\mathcal{N}_i, \mathcal{N}_j)$. On emploiera comme similarité $S\left(\mathcal{N}_i, \mathcal{N}_j\right) = K(\mathcal{N}_i, \mathcal{N}_j)$. On a alors (résultat admis) :

$$\log K\left(\mathcal{N}_i, \mathcal{N}_j\right) = -\frac{1}{2}\log\left|\Sigma_i + \Sigma_j\right| - \frac{1}{2}\left(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right)^T \left(\Sigma_i + \Sigma_j\right)^{-1} \left(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right) - \frac{n}{2}\log 2\pi$$

Où $|\Sigma|$ est le déterminant de Σ.

Calculer $\frac{\partial \log K(\mathcal{N}_i, \mathcal{N}_j)}{\partial \boldsymbol{\mu}_i}$ et $\frac{\partial \log K(\mathcal{N}_i, \mathcal{N}_j)}{\partial \Sigma_i}$ (pour la première on peut utiliser la dérivée d'un scalaire par rapport à un vecteur, pour la seconde, on passera par la dérivée de la fonction scalaire par rapport aux composantes Σ_{kl} de Σ).

3. Similarité asymétrique : Distance de Kulback-Leibler

La deuxième alternative est la distance de Kulback-Leibler entre deux distributions : $D_{KL}(\mathcal{N}_j || \mathcal{N}_i) = \int_z \left(\log \frac{\mathcal{N}_j(\boldsymbol{z})}{\mathcal{N}_i(\boldsymbol{z})}\right) \mathcal{N}_j(\boldsymbol{z}) d\boldsymbol{z}$, on emploiera comme similarité $S\left(\mathcal{N}_i, \mathcal{N}_j\right) = -D_{KL}(\mathcal{N}_j || \mathcal{N}_i)$

3.1 Montrer :

$$\int_z \left(\log \mathcal{N}_i(\boldsymbol{z})\right) \mathcal{N}_j(\boldsymbol{z}) d\boldsymbol{z} = -\frac{1}{2}\left(n \log 2\pi + \log|\Sigma_i| + Tr\left(\Sigma_i^{-1}\Sigma_j\right) + \left(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right)^T \Sigma_i^{-1} \left(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right)\right)$$

3.2 En déduire :

$$D_{KL}\left(\mathcal{N}_j, \mathcal{N}_i\right) = \frac{1}{2}\left(Tr\left(\Sigma_i^{-1}\Sigma_j\right) + \left(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right)^T \Sigma_i^{-1} \left(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right) - n - \log \frac{|\Sigma_j|}{|\Sigma_i|}\right)$$

3.3 Calculer $\frac{\partial D_{KL}(\mathcal{N}_j, \mathcal{N}_i)}{\partial \mu_i}$, $\frac{\partial D_{KL}(\mathcal{N}_j, \mathcal{N}_i)}{\partial \Sigma_i}$, $\frac{\partial D_{KL}(\mathcal{N}_j, \mathcal{N}_i)}{\partial \Sigma_j}$

4. Algorithme d'apprentissage. Etant donné un terme w et un terme de contexte c, on va optimiser par gradient stochastique une fonction de coût qui a la forme suivante :

$L\left(w, c_p, c_n\right) = \max\left(0, 1 - S\left(w, c_p\right) + S(w, c_n)\right)$ où c_p est un terme de contexte « positif », i.e. qui co-occurre avec w dans le corpus et c_n est un terme de contexte négatif. $S(,)$ dénote l'une des deux similarités introduites ci-dessus et par abus de notation $S(w, c)$ désigne la similarité des représentations correspondantes.

4.1 Expliquer cette fonction de coût.

4.2 Donner un algorithme de gradient stochastique pour chacune des deux similarités, qui optimise cette fonction de coût.

Réseaux convolutionnels

On a vu en cours les auto-encodeurs avec des connexions complètes. On veut construire un auto-encodeur convolutionnel : un réseau avec une seule couche cachée dont l'encodeur et le décodeur sont tous les deux des couches convolutionnelles. On examine une solution possible. On considère une image X, sa convolution avec un filtre W donne une image transformée Z

$X =$ | x_1 | x_2 | x_3 |
|---|---|---|
| x_4 | x_5 | x_6 |
| x_7 | x_8 | x_9 |

$W =$ | w_1 | w_2 |
|---|---|
| w_3 | w_4 |

$Z =$ | z_1 | z_2 |
|---|---|
| z_3 | z_4 |

1. Exprimer chacun des z_i en fonction des x_j et des w_k
2. On aplatit ces matrices : on transforme les matrices X et Z sous la forme de vecteurs en les parcourant de gauche à droite et de haut en bas et. Par exemple Z devient $\boldsymbol{z} = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix}$. Ecrire la convolution de X par W (l'encodeur) sous une forme matricielle : $\boldsymbol{z} = \boldsymbol{w}\boldsymbol{x}$ où $\boldsymbol{w}$ est une matrice de taille appropriée.
3. Donner l'expression de $\boldsymbol{y} = \boldsymbol{w}^T\boldsymbol{z}$. On parle de convolution transposée.
4. En quoi cette expression permet-elle de construire un auto-encodeur convolutionnel ?
5. On peut implémenter le décodeur $\boldsymbol{y} = \boldsymbol{w}^T\boldsymbol{z}$ sous la forme d'une convolution. Pour cela on considère la matrice Z' où l'on a ajouté à Z des cellules de « padding » qui sont mises à 0. Montrer par un dessin comment le décodeur peut être implémenté sous forme de convolution de Z' avec un noyau 2x2 et un stride de 1. Donner le noyau W' correspondant.

Z'= | | | | |
|---|---|---|---|
| | z_1 | z_2 | |
| | z_3 | z_4 | |
| | | | |

6. On veut implémenter un autoencodeur convolutionnel sur une image X, de dimensions 4x4, avec un noyau 2x2 comme ci-dessus, mais avec un déplacement (stride) de 2. On obtient là aussi un codage Z de taille 2x2. X est représentée par la matrice ci-dessous, W et Z sont inchangés.

X= | x_1 | x_2 | x_3 | x_4 |
|---|---|---|---|
| x_5 | x_6 | x_7 | x_8 |
| x_9 | x_{10} | x_{11} | x_{12} |
| x_{13} | x_{14} | x_{15} | x_{16} |

6.1 Quelle est la matrice $\boldsymbol{w}$ permettant de coder cette convolution sous forme matricielle ?

Donner par un dessin l'implémentation du décodeur sous la forme d'une convolution du codage Z, avec un noyau 2x2 et stride de 1.

Formules utiles

A, B et X sont des matrices qui ont les bonnes dimensions pour se multiplier, x est un scalaire, $Tr(A)$ est la trace de A et $|A|$ son déterminant, $\boldsymbol{x}$ est un vecteur, x est un scalaire.

Trace de matrices

$Tr(A+B) = Tr(A) + Tr(B)$, $Tr(AB) = Tr(BA)$, $x^T x = Tr(xx^T)$, $x^T Ax = Tr(xAx^T) = Tr(Axx^T)$

Dérivation matricielle

La dérivée d'un scalaire par rapport à un vecteur est le vecteur des dérivées du scalaire par rapport à chacune des composantes (le gradient habituel) $\left[\frac{\partial y}{\partial x_1}, \dots, \frac{\partial y}{\partial x_n}\right]$. De même la dérivée d'un scalaire par rapport à une matrice est la matrice des dérivées de ce scalaire par rapport à chaque élément de la matrice $\begin{pmatrix} \frac{\partial y}{\partial x_{11}} & \cdots & \frac{\partial y}{\partial x_{p1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial y}{\partial x_{1q}} & \cdots & \frac{\partial y}{\partial x_{pq}} \end{pmatrix}$.

La dérivée d'un vecteur par rapport à un scalaire est le vecteur des dérivées des composantes par rapport à ce scalaire $\left[\frac{\partial y_1}{\partial x}, \dots, \frac{\partial y_n}{\partial x}\right]^T$. La dérivée d'une matrice par rapport à un scalaire est la matrice des dérivées des composantes $\begin{pmatrix} \frac{\partial y_{11}}{\partial x} & \cdots & \frac{\partial y_{1n}}{\partial x} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_{m1}}{\partial x} & \cdots & \frac{\partial y_{mn}}{\partial x} \end{pmatrix}$.

Dérivées de scalaires

On suppose que le scalaire dépend des coefficients de la matrice ou l'inverse suivant les cas.

$\frac{\partial}{\partial A} Tr(AB) = B^T$, $\frac{\partial}{\partial A} \ln|A| = (A^{-1})^T = (A^T)^{-1}$, $\frac{\partial \ln|A|}{\partial x} = Tr(A^{-1}\frac{\partial A}{\partial x})$, $\frac{\partial Tr(A)}{\partial x} = Tr(\frac{\partial A}{\partial x})$, $\frac{\partial \boldsymbol{x}^T A \boldsymbol{x}}{\partial \boldsymbol{x}} = (A + A^T)\boldsymbol{x}$, $\frac{\partial \boldsymbol{a}^T \boldsymbol{x}}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{x}^T \boldsymbol{a}}{\partial \boldsymbol{x}} = \boldsymbol{a}$, $\frac{\partial \boldsymbol{x}^T A^{-1} \boldsymbol{y}}{\partial \boldsymbol{A}} = -(A^{-1})^T \boldsymbol{x}\boldsymbol{y}^T (A^{-1})^T$, $\frac{\partial}{\partial A} Tr(X^T A^{-1} Y) = -(A^{-1} Y X^T A^{-1})^T$

Dérivées de matrices par rapport à un scalaire

On suppose que les coefficients de la matrice dépendent su scalaire.

$\frac{\partial A^{-1}}{\partial x} = -A^{-1}\frac{\partial A}{\partial x}A^{-1}$, $\frac{\partial AB}{\partial x} = \frac{\partial A}{\partial x}B + A\frac{\partial B}{\partial x}$

Espérance et trace

$E_x[A\boldsymbol{x}] = AE_x[\boldsymbol{x}]$, $E_x[Tr(A\boldsymbol{x})] = Tr(AE_x(\boldsymbol{x})]$