Modèles génératifs pour les données textuelles

Vecteurs de mots

(4 points) Exercice log-vraisemblance et vecteurs de mots

Soit la phrase $s =$ *le petit chat dort*. On veut entraîner un modèle jouet `word2vec` par maximisation de la vraisemblance sur un corpus contenant uniquement la phrase s. Pour simplifier les calculs on considèrera que le voisinage est de taille 1, et que les vecteurs (contextes et centres) sont de taille 1.

1. Combien y a-t-il de paramètres dans ce modèle ?

2. On suppose que tous les paramètres sont initialisés à la valeur 1, et que le pas de descente de gradient vaut $0,1$. Donner les valeurs des paramètres après la première mise à jour des paramètres. Décrivez bien les calculs.

Modèles de langue

(4 points) Exercice : Modèles n-grammes et Évaluation

1. Considérez un modèle bigramme avec vocabulaire $\mathcal{V} = \{\texttt{<s>}, \texttt{a}, \texttt{b}, \texttt{</s>}\}$ et le corpus d'entraînement suivant:

```
<s> a b </s>
<s> a a </s>
<s> b a </s>
```

   Rappelez l'equation pour l'estimation de maximum de vraisemblance (MLE) et calculez la pour toutes les probabilités bigrammes $p(w_j|w_i)$ pour tous $w_i, w_j \in \mathcal{V}$. Présentez vos résultats dans un tableau 4×4.

2. En utilisant le modèle bigramme de la partie (a), calculez la probabilité de la séquence <s> a b a </s>.

3. Supposons qu'une nouvelle séquence <s> b b </s> soit observée. En utilisant le lissage de Laplace (add-one), calculez:

   - La probabilité lissée $p(\texttt{b}|\texttt{b})$
   - La probabilité lissée $p(\texttt{</s>}|\texttt{b})$
   - Que vaut $\sum_{w \in \mathcal{V}} p(w|\texttt{b})$ après lissage ?

4. Considérez deux modèles de langage, M_1 et M_2, avec les perplexités suivantes sur un ensemble de test:

$$\text{Perplexité}(M_1) = 120$$
$$\text{Perplexité}(M_2) = 60$$

   Calculez la différence en bits par token entre ces modèles. Lequel des deux modèles est le meilleur selon cette métrique ?

(2 points) Exercice : Modèles de langage neuronaux

1. Considérez un modèle de langage neuronal feed-forward avec :

   - Taille du vocabulaire $|\mathcal{V}| = 10,000$
   - Dimension des plongements lexicaux $d = 300$
   - Taille de la couche cachée $m = 500$
   - Longueur du contexte de 3 mots

   Calculez le nombre total de paramètres dans ce modèle. Détaillez votre calcul en montrant le nombre de paramètres pour chaque composant.

Architectures neuronales

(4) Exercice : Architecture du Transformer

Calculez la sortie d'un Transformer pour une séquence simple. Soit une séquence d'entrée de 2 tokens avec des embeddings:

$$\mathbf{X} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{2\times 2}$$

Ce Transformer comporte une seule couche avec:

- **Mécanisme d'attention**: Matrices de projection $\mathbf{W}^Q = \mathbf{W}^K = \mathbf{W}^V = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
- **Masque d'attention**: $\mathbf{M} = \begin{bmatrix} 0 & -\infty \\ 0 & 0 \end{bmatrix}$ (causal)
- **Réseau feed-forward**: $\mathbf{W}_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$, $\mathbf{b}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $\mathbf{W}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$, $\mathbf{b}_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$
- **Normalisation**: Supposez $\text{LayerNorm}(\mathbf{x}) = \mathbf{x}$ pour simplifier
- **Projection finale**: $\mathbf{W}_{\text{out}} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$, $\mathbf{b}_{\text{out}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

Calculez pas à pas:

1. Les matrices $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$
2. La matrice d'attention avec masque et les scores après softmax
3. La sortie de la couche d'attention puis de la couche feed-forward (avec les connexions résiduelles)
4. Les logits finaux et les probabilités pour les deux positions

(4 points) Exercice : Des parenthèses

Considérez le problème de la détection des parenthèses bien équilibrées. Vous devez déterminer si chaque parenthèse ouvrante '(' a une parenthèse fermante ')' correspondante, et si elles sont correctement imbriquées. Par exemple:

- "(())", "((()))" sont valides
- "(()", ")()" et "())()" sont invalides

1. Montrez pourquoi un RNN simple peut théoriquement résoudre ce problème alors qu'un modèle n-gramme (même avec n grand) ne le peut pas. Donnez une spécification précise de :
   - La dimension minimale nécessaire pour l'état caché $\mathbf{h}_t$
   - Comment initialiser l'état caché $\mathbf{h}_0$
   - La transformation exacte que $\mathbf{h}_t$ doit subir à chaque étape
   - Comment classifier la séquence comme valide ou invalide à partir de l'état final
2. Montrez comment un mécanisme d'attention peut être utilisé pour résoudre ce problème. Spécifiquement:
   - Donnez une représentation des tokens d'entrée (parenthèses ouvrantes et fermantes) qui permet de résoudre ce problème.
   - Concevez un mécanisme d'attention avec une seule tête qui peut apprendre à associer chaque parenthèse fermante avec sa correspondante ouvrante.
   - Expliquez pourquoi les encodages positionnels sont nécessaires ou non pour cette tâche.
   - Proposez une méthode pour extraire la décision finale (valide/invalide) à partir des sorties du mécanisme d'attention.

RLHF

(5 points) Exercice Gradient des pertes RLHF

Dans cet exercice, on demande le gradient des fonctions de pertes pour RLHF.

1. On a vu que le modèle de récompenses définissait une fonction de perte pour un triplet (x, y_c, y_r) et un réseau de neurones r_ϕ:

   $$L(\phi; y_c; y_r) = \log \Big(1 + \exp\big(r_\phi(y_r) - r_\phi(y_c)\big)\Big)$$

   Donner le gradient de $\nabla_\phi L(\phi)$ en fonction des gradients de $r_\phi(y_c)$ et de $r_\phi(y_r)$.

2. Même question pour DPO. On rappelle que la fonction de perte est donnée par:

   $$L(\phi; y_c, y_r) = -\log \sigma\big(r_\phi(y_c) - r_\phi(y_r)\big)$$

   où σ est la fonction sigmoïde, $\sigma(x) = \frac{1}{1+\exp(-x)}$.
   Décrivez bien les étapes de calcul.

3. Que remarque-t'on ?