

Série d'exercices : IA Générative et LLMs

Exercice 1 : Concepts Fondamentaux

Sujet : IA Discriminative vs. Générative

Question : Un ingénieur développe deux modèles d'IA :

1. **Modèle A :** Un système qui lit un e-mail et lui attribue une étiquette : "Important", "Spam" ou "Standard".
2. **Modèle B :** Un système qui lit le même e-mail et rédige une proposition de réponse en une phrase.

Pour chaque modèle, déterminez s'il est **discriminatif** ou **génératif**. Justifiez chaque choix en une phrase, en utilisant les concepts de "classification" vs "création" ou "apprendre à juger" vs "apprendre l'essence".

Exercice 2 : Anatomie des Embeddings

Sujet : Matrice d'Embedding et Espace Sémantique

Question : Imaginons un petit LLM "jouet" avec les caractéristiques suivantes :

- Taille du vocabulaire : 10 000 tokens
 - Dimension de l'embedding (d_{model}) : 128
1. Combien de paramètres (poids) la matrice d'embedding de ce modèle contient-elle au total ? (Montrez votre calcul).
 2. Après l'entraînement, on observe l'analogie vectorielle suivante :

$$\vec{v}(\text{"Madrid"}) - \vec{v}(\text{"Espagne"}) + \vec{v}(\text{"France"}) \approx \vec{v}(\text{"Paris"})$$

En vous basant sur le cours, qu'est-ce que cette équation nous apprend sur la manière dont le modèle a organisé l'"essence" de ces concepts dans son espace sémantique ?

Exercice 3 : Calcul d'Attention

Sujet : Mécanisme d'Attention (Q, K, V)

Question : Considérons une étape simplifiée du calcul d'attention pour le mot T_1 ("Le") dans la phrase "Le chat dort". Le modèle doit calculer son affinité avec le mot T_2 ("chat"). Après projection, nous avons les vecteurs Q_1 (Requête de "Le") et K_2 (Clé de "chat").

- $Q_1 = [1.0, 0.5]$
- $K_2 = [0.8, -0.4]$
- La dimension des vecteurs est $d_k = 2$.

1. Calculez le **score d'affinité brut** $S_{1,2}$ (l'affinité de "Le" pour "chat") en utilisant la formule :

$$S_{ij} = \frac{Q_i \cdot K_j}{\sqrt{d_k}}$$

2. Le score calculé est-il élevé ou bas ? Si, après ce calcul, le score $S_{1,1}$ ("Le" pour "Le") était de 0.1 et le score $S_{1,2}$ ("Le" pour "chat") était de 5.0, qu'est-ce que cela signifierait pour le modèle lors de la mise à jour de l'embedding de "Le" ?

Exercice 4 : Rôles des Couches

Sujet : Multi-Head Attention vs. FFN

Question : Un bloc décodeur de Transformer est composé de deux sous-couches principales : l'**Attention Multi-têtes** et le **Réseau Feed-Forward (FFN)**. Ces deux couches modifient les vecteurs de tokens, mais leurs rôles sont différents.

1. Expliquez la différence fondamentale de leur fonction : comment l'Attention Multi-têtes gère-t-elle l'**interaction entre les tokens**, et que fait le FFN pour **chaque token individuellement** ?
2. Le cours utilise l'analogie d'une "salle de brainstorming" pour le FFN, où le vecteur est d'abord projeté dans un espace de plus grande dimension (expansion) avant d'être ramené à sa taille d'origine (contraction). Pourquoi cette "expansion" temporaire est-elle cruciale pour la puissance du modèle ?

Exercice 5 : Processus d'Apprentissage

Sujet : Loss et Backpropagation

Question : Un LLM est entraîné sur la phrase "Le soleil brille...". La **vérité terrain** (le mot correct suivant) est "dans" (ID 80). Le modèle effectue un **Forward Pass** et sa couche Softmax finale produit les probabilités suivantes pour les tokens les plus probables :

- $P(\text{"sur"})$ (ID 75) = 0.50
- $P(\text{"dans"})$ (ID 80) = 0.10
- $P(\text{"le"})$ (ID 04) = 0.30
- $P(\text{autres tokens})$ = 0.10

1. Calculez l'**Erreur (Loss)** pour cette prédiction en utilisant la formule de Cross-Entropy Loss :

$$Erreur = -\log(P(\text{mot correct}))$$

2. Pendant la **Rétropropagation** (Backward Pass), le modèle calculera des gradients. Le gradient indiquera-t-il aux poids du modèle d'*augmenter* ou *diminuer* la probabilité du token "sur" (ID 75) lors de la prochaine itération ?

3. Le **Learning Rate** (Taux d'Apprentissage) est un hyperparamètre crucial de l'Optimiseur. Si le Learning Rate était réglé trop *haut* (par exemple, à 1.0 au lieu de 0.0001), quel problème chaotique pourrait survenir pendant la mise à jour des poids, même si les gradients sont corrects ?

Exercice 6 : Paradoxe de l'Inférence

Sujet : Parallélisme vs. Séquentiel

Question : Le cours met en évidence un paradoxe :

- Pendant l'**entraînement**, le Transformer est massivement parallèle. Le FFN, par exemple, traite tous les tokens d'une phrase (ex : 1000 tokens) "exactement en même temps".
- Pendant la **génération** (inférence), le modèle est "auto-régressif" et doit générer les mots "un par un", ce qui est beaucoup plus lent.

Expliquez ce paradoxe en répondant à deux questions :

1. Qu'est-ce qui **permet** au FFN de traiter tous les tokens en parallèle pendant l'entraînement ? (Indice : de quoi dépend le calcul FFN pour le mot "chat" ?)
2. Qu'est-ce qui **empêche** le modèle de générer les 100 mots d'une réponse en parallèle pendant l'inférence ? (Indice : que doit *connaître* le modèle pour prédire le mot $N + 1$?)