Date remise: Lundi 14 novembre, 23h

Instructions

- *Montrez votre démarche pour toutes les questions !*
- *Utilisez LaTeX et le modèle que nous vous fournissons pour rédiger vos réponses. Vous pouvez réutiliser la plupart des raccourcis de notation, des équations et/ou des tableaux. SVP voir la politique des devoirs sur le site web du cours pour plus de détails.*
- *Vous devez soumettre toutes vos réponses sur la page Gradescope du cours*
- *Les TAs pour ce devoir sont* ***Arian Khorasani et Nanda Harishankar Krishna***

Question 1 (2-5-5-5-3)**.** (**Rétropropagation du gradient dans Réseau de neurones récurrents**)

Considérez l'unité récurrente suivante:

$$\boldsymbol{h}_t = \sigma(\boldsymbol{W}\boldsymbol{x}_t + \boldsymbol{U}\boldsymbol{h}_{t-1})$$
$$\boldsymbol{y}_t = \boldsymbol{V}\boldsymbol{h}_t$$

oú σ denotes the logistic sigmoid function. Que $\boldsymbol{z}_t$ soit la vraie cible de la prédiction $\boldsymbol{y}_t$ et considère la somme des pertes au carré $L = \sum_t L_t$ oú $L_t = ||\boldsymbol{z}_t - \boldsymbol{y}_t||_2^2$.

Dans cette question, notre but est d'obtenir une expression pour les gradients $\nabla_{\boldsymbol{W}} L$ and $\nabla_{\boldsymbol{U}} L$.

1. Tout d'abord, remplissez le graphique de calcul suivant pour ce l'unité récurrente, déroulé pendant 3 étapes (de $t = 1$ à $t = 3$). Étiqueter chaque nœud avec l'unité cachée correspondante et chaque bord avec le poids correspondant.

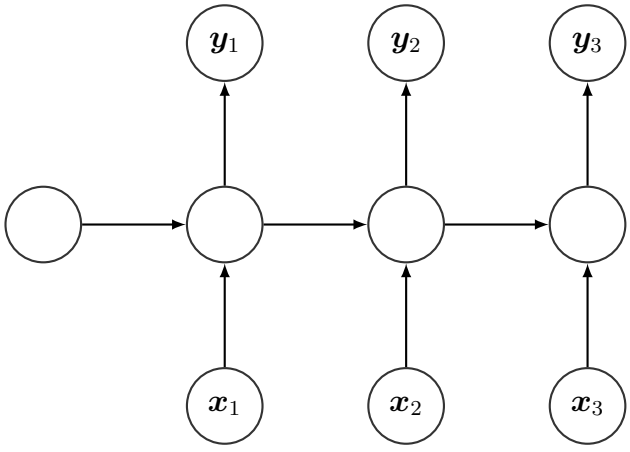

FIGURE 1 – Graphique de calcul de l'unité récurrente déroulé en trois temps.

2. Dériver une expression récursive pour le gradient total $\nabla_{\boldsymbol{h}_t} L$ en termes de $\nabla_{\boldsymbol{h}_t} L_t$ et $\nabla_{\boldsymbol{h}_{t+1}} L$.

3. obtenir une expression pour $\nabla_{\boldsymbol{h}_t} L_t$ et $\frac{\partial \boldsymbol{h}_{t+1}}{\partial \boldsymbol{h}_t}$.
4. Dérivez maintenant $\nabla_{\boldsymbol{W}} L$ et $\nabla_{\boldsymbol{U}} L$ en fonction de $\nabla_{\boldsymbol{h}_t} L$, en utilisant les résultats des deux sous-questions précédentes.
   Indice: Il pourrait être utile de tenir compte de la contribution des matrices de poids lors du calcul de la unité cachée récurrente à un moment particulier t et comment ces contributions pourraient être agrégées.
5. Éviter le problème de l'explosion ou de la disparition des gradients, nous pouvons utiliser BPTT tronqué pour calculer les gradients pour les mises à jour des paramètres dans les RNNs. Que pouvez-vous dire au sujet des estimations des gradients du BPTT tronqué – sont-ils impartiaux ou biaisés ? Pourquoi vous le pensez ? Bien que le BPTT tronqué puisse aider à atténuer l'explosion ou la disparition des gradients, Quels problèmes entrevoyez-vous avec cette approche, par exemple dans les tâches impliquant la langue ?

Answer 1. Dans cette question, $\boldsymbol{U}$ est la matrice de poids récurrents, $\boldsymbol{W}$ est la matrice de poids d'entrée et $\boldsymbol{V}$ est la matrice de poids de sortie. Vous pouvez vous référer à ces liens pour les questions sur le gradient : le Deep Learning book (10.2.2) et le Matrix Cookbook. Notez que nous utilisons le **notation et disposition du calcul matriciel conformes au livre Deep Learning**, comme spécifié dans la politique de devoirs.

1. Voici la réponse dans Fig. 2:

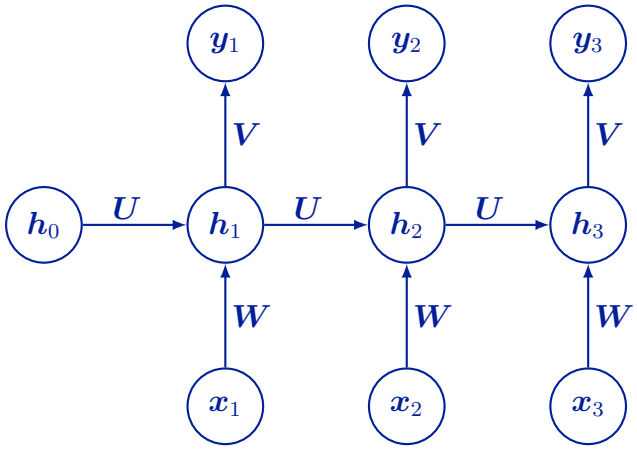

FIGURE 2 – Réponse à la Q1.1.

2. Soit T le dernier pas de temps, nous avons

$$\nabla_{\boldsymbol{h}_t} L = \nabla_{\boldsymbol{h}_t} \sum_{t'=0}^{T} L_{t'} = \frac{\partial L_0}{\partial \boldsymbol{h}_t} + \frac{\partial L_1}{\partial \boldsymbol{h}_t} + \cdots + \frac{\partial L_t}{\partial \boldsymbol{h}_t} + \frac{\partial L_{t+1}}{\partial \boldsymbol{h}_t} + \cdots + \frac{\partial L_{T-1}}{\partial \boldsymbol{h}_t} + \frac{\partial L_T}{\partial \boldsymbol{h}_t}$$

Nous savons que $\boldsymbol{h}_t$ influence la perte au t et au-delà car la sortie aux pas de temps précédents n'en depends pas. Donc, le gradient de la perte aux pas de temps précédant t par rapport à $\boldsymbol{h}_t$ est 0. Donc, nous avons

$$\nabla_{\boldsymbol{h}_t} L = \cancel{\frac{\partial L_0}{\partial \boldsymbol{h}_t}} + \cancel{\frac{\partial L_1}{\partial \boldsymbol{h}_t}} + \cdots + \frac{\partial L_t}{\partial \boldsymbol{h}_t} + \frac{\partial L_{t+1}}{\partial \boldsymbol{h}_t} + \cdots + \frac{\partial L_{T-1}}{\partial \boldsymbol{h}_t} + \frac{\partial L_T}{\partial \boldsymbol{h}_t}$$
$$= \nabla_{\boldsymbol{h}_t} L_t + \frac{\partial L_{t+1}}{\partial \boldsymbol{h}_t} + \cdots + \frac{\partial L_{T-1}}{\partial \boldsymbol{h}_t} + \frac{\partial L_T}{\partial \boldsymbol{h}_t}$$

De la même manière, nous avons

$$\nabla_{\boldsymbol{h}_{t+1}} L = \frac{\partial L_{t+1}}{\partial \boldsymbol{h}_{t+1}} + \cdots + \frac{\partial L_T}{\partial \boldsymbol{h}_{t+1}}$$

En utilisant ceci dans la résultat précédant, nous avons

$$\nabla_{\boldsymbol{h}_t} L = \nabla_{\boldsymbol{h}_t} L_t + \left(\frac{\partial \boldsymbol{h}_{t+1}}{\partial \boldsymbol{h}_t}\right)^\top \left(\frac{\partial L_{t+1}}{\partial \boldsymbol{h}_{t+1}} + \cdots + \frac{\partial L_T}{\partial \boldsymbol{h}_{t+1}}\right) = \boxed{\nabla_{\boldsymbol{h}_t} L_t + \left(\frac{\partial \boldsymbol{h}_{t+1}}{\partial \boldsymbol{h}_t}\right)^\top \nabla_{\boldsymbol{h}_{t+1}} L}$$

Note: vous devez prendre la transposition de $\frac{\partial \boldsymbol{h}_{t+1}}{\partial \boldsymbol{h}_t}$. Intuition:

$$\nabla_{\boldsymbol{h}_t} L_{t+1} = \begin{bmatrix} \frac{\partial L_{t+1}}{\partial (\boldsymbol{h}_t)_1} \\ \vdots \\ \frac{\partial L_{t+1}}{\partial (\boldsymbol{h}_t)_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial L_{t+1}}{\partial (\boldsymbol{h}_{t+1})_1}\frac{\partial (\boldsymbol{h}_{t+1})_1}{\partial (\boldsymbol{h}_t)_1} + \cdots + \frac{\partial L_{t+1}}{\partial (\boldsymbol{h}_{t+1})_n}\frac{\partial (\boldsymbol{h}_{t+1})_n}{\partial (\boldsymbol{h}_t)_1} \\ \vdots \\ \frac{\partial L_{t+1}}{\partial (\boldsymbol{h}_{t+1})_1}\frac{\partial (\boldsymbol{h}_{t+1})_1}{\partial (\boldsymbol{h}_t)_n} + \cdots + \frac{\partial L_{t+1}}{\partial (\boldsymbol{h}_{t+1})_n}\frac{\partial (\boldsymbol{h}_{t+1})_n}{\partial (\boldsymbol{h}_t)_n} \end{bmatrix}$$

$$\frac{\partial \boldsymbol{h}_{t+1}}{\partial \boldsymbol{h}_t} = \begin{bmatrix} \frac{\partial (\boldsymbol{h}_{t+1})_1}{\partial (\boldsymbol{h}_t)_1} & \cdots & \frac{\partial (\boldsymbol{h}_{t+1})_1}{\partial (\boldsymbol{h}_t)_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial (\boldsymbol{h}_{t+1})_n}{\partial (\boldsymbol{h}_t)_1} & \cdots & \frac{\partial (\boldsymbol{h}_{t+1})_n}{\partial (\boldsymbol{h}_t)_n} \end{bmatrix} \quad \text{and} \quad \nabla_{\boldsymbol{h}_{t+1}} L_{t+1} = \begin{bmatrix} \frac{\partial L_{t+1}}{\partial (\boldsymbol{h}_{t+1})_1} \\ \vdots \\ \frac{\partial L_{t+1}}{\partial (\boldsymbol{h}_{t+1})_n} \end{bmatrix}$$

$$\left(\frac{\partial \boldsymbol{h}_{t+1}}{\partial \boldsymbol{h}_t}\right)^\top \nabla_{\boldsymbol{h}_{t+1}} L_{t+1} = \begin{bmatrix} \frac{\partial (\boldsymbol{h}_{t+1})_1}{\partial (\boldsymbol{h}_t)_1} & \cdots & \frac{\partial (\boldsymbol{h}_{t+1})_n}{\partial (\boldsymbol{h}_t)_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial (\boldsymbol{h}_{t+1})_1}{\partial (\boldsymbol{h}_t)_n} & \cdots & \frac{\partial (\boldsymbol{h}_{t+1})_n}{\partial (\boldsymbol{h}_t)_n} \end{bmatrix} \begin{bmatrix} \frac{\partial L_{t+1}}{\partial (\boldsymbol{h}_{t+1})_1} \\ \vdots \\ \frac{\partial L_{t+1}}{\partial (\boldsymbol{h}_{t+1})_n} \end{bmatrix} = \nabla_{\boldsymbol{h}_t} L_{t+1}$$

3. Nous avons

$$\nabla_{\boldsymbol{h}_t} L_t = \nabla_{\boldsymbol{h}_t} \|\boldsymbol{z}_t - \boldsymbol{y}_t\|_2^2 = \nabla_{\boldsymbol{h}_t} \|\boldsymbol{z}_t - \boldsymbol{V}\boldsymbol{h}_t\|_2^2 = \boxed{-2\boldsymbol{V}^\top (\boldsymbol{z}_t - \boldsymbol{V}\boldsymbol{h}_t)} = \boxed{-2\boldsymbol{V}^\top (\boldsymbol{z}_t - \boldsymbol{y}_t)}$$

$$\frac{\partial \boldsymbol{h}_{t+1}}{\partial \boldsymbol{h}_t} = \frac{\partial}{\partial \boldsymbol{h}_t} \sigma(\boldsymbol{W}\boldsymbol{x}_{t+1} + \boldsymbol{U}\boldsymbol{h}_t)$$
$$= \operatorname{diag}(\sigma(\boldsymbol{W}\boldsymbol{x}_{t+1} + \boldsymbol{U}\boldsymbol{h}_t) \odot (1 - \sigma(\boldsymbol{W}\boldsymbol{x}_{t+1} + \boldsymbol{U}\boldsymbol{h}_t)))\boldsymbol{U}$$
$$= \boxed{\operatorname{diag}(\boldsymbol{h}_{t+1} \odot (1 - \boldsymbol{h}_{t+1}))\boldsymbol{U}}$$

4. Nous avons

$$\nabla_{\boldsymbol{W}} L = \sum_t \sum_i \frac{\partial L}{\partial (\boldsymbol{h}_t)_i} \frac{\partial (\boldsymbol{h}_t)_i}{\partial \boldsymbol{W}} = \sum_t \operatorname{diag}(\boldsymbol{h}_t \odot (1 - \boldsymbol{h}_t))(\nabla_{\boldsymbol{h}_t} L)\boldsymbol{x}_t^\top$$
$$\nabla_{\boldsymbol{U}} L = \sum_t \sum_i \frac{\partial L}{\partial (\boldsymbol{h}_t)_i} \frac{\partial (\boldsymbol{h}_t)_i}{\partial \boldsymbol{U}} = \sum_t \operatorname{diag}(\boldsymbol{h}_t \odot (1 - \boldsymbol{h}_t))(\nabla_{\boldsymbol{h}_t} L)\boldsymbol{h}_{t-1}^\top$$

5. Le BPTT tronqué fournit des estimations biasées du gradient en raison de la courte longeur des étapes et de la réinitialisation du calcul du gradient après ces étapes. Donc, il se rapproche du gradient. Ainsi, il n'est pas garanti qu'il converge vers des minima. Il peut ne pas être bien adaptée aux tâches où les dépendances à long-terme sont importantes.

Question 2. (**Transformateurs sont GNNs**)

L'apprentissage des représentations des intrants est le socle de tous les réseaux neuronaux. Au cours des dernières années, Les modèles de transformateurs ont été largement adaptés aux tâches de modélisation de séquence dans la vision et domaines de langue, tandis que les réseaux neuronaux graphiques (GNN) ont été efficaces dans la construction de représentations de nœuds et les bords dans les données graphiques. Dans les questions suivantes, nous allons explorer les Transformers et les GNN, et dessiner quelques connexions entre eux.

Contexte:

Examinons d'abord un modèle graphique. Nous définissons un graphique dirigé $G = \{V, E\}$ où V est l'ensemble de tous les sommets et E est l'ensemble de tous les bords. pour $\forall v_i \in V$, laissez-nous définir $\mathcal{N}(v_i)$ comme ensemble de tous les voisins de v_i avec des bords sortants vers v_i. v_i a une représentation d'état h_i^t à chaque étape de temps t.

Les valeurs de h_i^t sont mis à jour en parallèle, en utilisant le même instantané du graphique à un pas de temps donné. Les procédures sont les suivantes : Nous devons d'abord agréger les données entrantes $H'_{it} = \{f_{ji}(h_j^t) | \forall j, v_j \in \mathcal{N}(v_i)\}$ de voisins utilisant la fonction $Agg(H'_{it})$. Notez que les données entrantes de chaque voisin est une version transformée de sa représentation en utilisant fonction f_{ji}. la fonction d'agrégation $Agg(H'_{it})$ peut être quelque chose comme la somme ou la moyenne des éléments dans H'_{it}.

Disons que l'état initial à l'étape 0 est h_i^0. Définissons maintenant la règle de mise à jour pour h_i^t à un pas de temps $t+1$ comme suit:

$$h_i^{t+1} = q(h_i^t, Agg(H'_{it})) \tag{1}$$

où q est un fonction – $Q_t : \{H_t, Agg(H'_t)\} \to H_t$, où $H_t = \{h_n^t | \forall n, v_n \in V\}$.

Maintenant, jetons un coup d'œil aux modèles Transformer. Rappelons que les modèles Transformer construisent des caractéristiques pour chaque mot en fonction des caractéristiques de tous les autres mots avec un mécanisme d'attention sur eux, tandis que les RNNs mettent à jour les fonctionnalités de manière séquentielle.

Pour représenter un mécanisme d'attention du modèle Transformer, définissons une représentation de caractéristiques h_i pour le mot i dans la phrase S. Nous avons l'équation standard pour la mise à jour de l'attention à la couche l en fonction de l'autre mot j comme suit:

$$h_i^{l+1} = \texttt{Attention}(Q^l h_i^l, K^l h_j^l, V^l h_j^l) \tag{2}$$

$$= \sum_{j \in S} \left(\texttt{softmax}_j(Q^l h_i^l \cdot K^l h_j^l) V^l h_j^l\right) \tag{3}$$

Où Q^l, K^l, V^l sont des matrices de poids pour « Query », « Key » et « Value ». Q est une matrice qui contient des représentations vectorielles d'un mot dans la phrase, Tandis que K est une matrice contenant des représentations pour tous les mots de la phrase. V est une autre matrice similaire à K qui a des représentations pour tous les mots de la phrase. Pour vous rafraîchir la mémoire au sujet du modèle de transformateur, vous pouvez vous reporter à paper.

En vous fondant sur les renseignements de base ci-dessus, répondez aux questions suivantes :

(2.1) Si l'opération d'agrégation pour $Agg(H'_{it})$ est la somme de la représentation de tous les sommets adjacents, Réécrire l'équation 1 en remplaçant $Agg(H'_{it})$) en termes de $\mathcal{N}$, f, et h.

(2.2) Considérons le graphique G de la figure 3. Les valeurs des sommets à l'étape t sont les suivantes

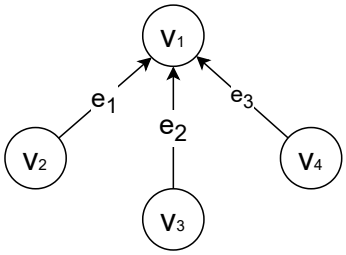

FIGURE 3 – Graphique G pour Q2.2

:

$$h_1^t = [1, -1] \quad h_2^t = [-1, 1] \quad h_3^t = [0, -1] \quad h_4^t = [1, 0] \tag{4}$$

La fonction d'agrégation $Agg(H'_{1t})$:

$$Agg(H'_{1t}) = [0.6, 0.2, 0.2] \begin{bmatrix} f(h_2^t) \\ f(h_3^t) \\ f(h_4^t) \end{bmatrix} \tag{5}$$

Et la fonction f sur tous les bords est:

$$f(x) = 2x \tag{6}$$

Maintenant, étant donné que

$$h_1^{t+1} = q(h_1^t, Agg(H'_{1t})) = W(h_1^t)^T + max\{Agg(H'_{1t}), 0\} \tag{7}$$

où $W = [1, 1]$, quelle est la valeur actualisée de h_1^{t+1} ?

(2.3) Considérons le graphique G en question (2.2). Nous voulons la modifier pour représenter la peine "Je mange des pommes rouges" (4 mots tokens) comme un graphique entièrement connecté. Chaque sommet représente un mot token, et les bords représentent les relations entre les tokens. Combien de bords au total le graphique G contient-il ? Notez que les bords sont orientés et un bord bidirectionnel compte comme deux bords.

(2.4) Au moyen des équations 1 et 3, Le mécanisme d'attention à tête unique du modèle de transformateur est équivalent à un cas particulier de GNN.

(2.5) Un domaine de recherche continu dans les modèles de transformateurs pour la NLP est le défi de l'apprentissage les dépendances à très long terme entre les entités dans une séquence. Compte tenu de ce lien avec les GNNs, pourquoi pensez-vous que cela pourrait être problématique ?

Answer 2. 1. Pour cela, nous agrégeons simplement tous les éléments de H'_{it} par sommation.

$$h_i^{t+1} = q\left(h_i^t, \sum_{j \in \mathcal{N}(i)} f_{ji}(h_j^t)\right)$$

2. $[0, 0.8]$

$$\begin{aligned} Wh_1^t &= [1,1][1,-1]^\top = 0 \\ Agg(h_1^t) &= Agg(\begin{bmatrix} 2h_2^t \\ 2h_3^t \\ 2h_4^t \end{bmatrix}) \\ &= [0.6, 0.2, 0.2] \begin{bmatrix} -2 & 2 \\ 0 & -2 \\ 2 & 0 \end{bmatrix} \\ &= [-0.8, 0.8] \\ q(h_1^t, Agg(H'_{1t})) &= [0,0] + max\{[-0.8, 0.8], 0\} = \boxed{[0,\ 0.8]} \end{aligned}$$

3. 12. Transformer modélise un graphe entièrement connecté parmi toutes les paires de mots. Un graphe entièrement connecté de 4 sommets a 12 arêtes dirigées.

4. Soit

$$\begin{aligned} f_{ji}(h_j^l) &= softmax_j(Q^l h_i^l \cdot (K^l h_j^l))V^l h_j^l \\ Agg(H'_i) &= \sum_{j \in \mathcal{N}(i)} f_{ji}(h_j^l) \\ q(h_i^t, Agg(H'_i)) &= f_{ii} + Agg(h_i) \end{aligned}$$

5. Le nombre d'arêtes dans le GNN, ainsi que le nombre de poids dans le modèle Transformer, évolue de façon quadratique avec le nombre de mots. Les besoins en mémoire ainsi que le nombre de FLOPs pour calculer la matrice d'attention évoluent avec la taille de l'entrée comme $O(n^2)$. Une explication détaillée des défis posés par les dépendances à long terme dans les modèles Transformer est disponible ici.

Question 3 (6-2-6-6)**.** (**Optimization et Regularization**)

(3.1) La normalisation par lots (batch normalization), la normalisation des couches (layer normalization) et la normalisation des instances (instance normalization) impliquent le calcul de la moyenne $\boldsymbol{\mu}$ et la variance $\boldsymbol{\sigma^2}$ par rapport à différents sous-ensembles des dimensions du tenseur. Étant donné le tenseur 3D suivant, calculez les tenseurs de moyenne et de variance correspondants pour chaque technique de normalisation: $\boldsymbol{\mu}_{batch}$, $\boldsymbol{\mu}_{layer}$, $\boldsymbol{\mu}_{instance}$, $\boldsymbol{\sigma}^2_{batch}$, $\boldsymbol{\sigma}^2_{layer}$, and $\boldsymbol{\sigma}^2_{instance}$.

$$\left[\begin{bmatrix}4,3,2\\1,4,3\end{bmatrix},\begin{bmatrix}3,4,1\\4,2,2\end{bmatrix},\begin{bmatrix}3,2,3\\4,1,2\end{bmatrix},\begin{bmatrix}1,1,3\\4,1,2\end{bmatrix}\right]$$

La taille de ce tenseur est de 4 x 2 x 3, ce qui correspond à la taille du lot, au nombre de canaux et au nombre de caractéristiques respectivement.

(3.2) Alors que BatchNorm est très commun dans les modèles de vision par ordinateur, Il est nettement surpassé par LayerNorm dans les tâches de traitement du langage naturel. Quels pourraient être quelques problèmes avec l'utilisation de BatchNorm dans NLP ? Vous pouvez considérer une mise en en œuvre naïve de BatchNorm et aussi illustrer votre point avec un exemple si vous le souhaitez.

(3.3) Considérez un problème de régression linéaire avec les données d'entrée $\boldsymbol{X} \in \mathbb{R}^{n\times d}$, poids $\boldsymbol{w} \in \mathbb{R}^{d\times 1}$ et objectifs $\boldsymbol{y} \in \mathbb{R}^{n\times 1}$. Supposons que l'exclusion est appliquée à l'entrée (avec une probabilité $1-p$ de laisser tomber l'unité, c.-à-d. la régler à 0). Soit $\boldsymbol{R} \in \mathbb{R}^{n\times d}$ être le masque dropout tel que $\boldsymbol{R}_{ij} \sim \text{Bern}(p)$ est échantillonné i.i.d. de la distribution Bernoulli.

Pour une fonction de perte d'erreur au carré avec dropout, on a alors :

$$L(\boldsymbol{w}) = ||\boldsymbol{y} - (\boldsymbol{X} \odot \boldsymbol{R})\boldsymbol{w}||^2$$

Soit Γ être une matrice diagonale avec $\Gamma_{ii} = (\boldsymbol{X}^\top \boldsymbol{X})^{1/2}_{ii}$. Montrer que les *expectation (over $\boldsymbol{R}$)* de la fonction de perte peut être réécrit comme: $\mathbb{E}[L(\boldsymbol{w})] = ||\boldsymbol{y} - p\boldsymbol{X}\boldsymbol{w}||^2 + p(1-p)||\Gamma\boldsymbol{w}||^2$. *Indice: Notez que nous essayons de trouver l'attente sur un terme carré et d'utiliser* $\text{Var}(Z) = \mathbb{E}[Z^2] - \mathbb{E}[Z]^2$.

(3.4) Considérez un problème de régression linéaire avec un vecteur d-dimensionnel $\boldsymbol{x} \in \mathbb{R}^d$ comme entrée. et $y_i \in \mathbb{R}$ comme sortie. L'ensemble de données comprend n exemples de formation. $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)$. Soit $\boldsymbol{X}$ être la $n \times d$ matrice de données formée en plaçant les vecteurs de caractéristiques sur les lignes de cette matrice c.-à-d,

$$\boldsymbol{X}^T = \left[\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\right]$$

Soit $\boldsymbol{y}$ être un vecteur de colonne avec des éléments $y_1, y_2 ... y_n$. Nous opérerons dans le cadre où $d > n$, c'est-à-dire qu'il y a plus de dimensions que d'échantillons. Ainsi, le système linéaire suivant est sous-limité/sous-déterminé:

$$\boldsymbol{X}\boldsymbol{w} = \boldsymbol{y} \tag{8}$$

Pour éviter le cas dégénéré, nous supposerons que $\boldsymbol{y}$ réside dans l'envergure de $\boldsymbol{X}$, c.-à-d. que ce système linéaire a au moins une solution.

Maintenant, rappelez-vous que le problème d'optimisation dans la régression des moindres carrés est le suivant :

$$\min_{\boldsymbol{w}\in\mathbf{R}^{\mathbf{d}}} \quad f(\boldsymbol{w}) = \sum_{i=1}^{n} \underbrace{(y_i - \boldsymbol{w}^T \boldsymbol{x}_i)^2}_{\text{squared error on example } i} \tag{9}$$

Nous optimiserons (9) par descente en pente. Plus précisément, initialisons $\boldsymbol{w}^{(0)} = 0$. Et à plusieurs reprises faire un pas dans la direction de gradient négatif avec un pas-taille constant suffisamment petit η jusqu'à convergence.

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \eta \nabla f(\boldsymbol{w}^{(t)}) \tag{10}$$

Référons-nous à la solution trouvée par la descente en pente à la convergence comme $\boldsymbol{w}^{gd}$. Prouver que la solution trouvée par descente en pente pour les moindres carrés est égale aux résultats de ce qui suit *différent* problème d'optimisation:

$$\boldsymbol{w}^{gd} = \arg\min_{\boldsymbol{w}\in\mathbf{R}^{\mathbf{d}}} \quad ||\boldsymbol{w}||_2^2 \tag{11}$$

$$\text{such that} \quad \boldsymbol{X}\boldsymbol{w} = \boldsymbol{y} \tag{12}$$

Indice: Pour ce problème d'optimisation, vous devez travailler avec le Lagrangien $L(\boldsymbol{w}, \lambda)$, donnée par

$$L(\boldsymbol{w}, \lambda) = \|\boldsymbol{w}\|^2 + \lambda^\top (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})$$

Obtenir $\boldsymbol{w}^*$ et λ^* à partir des conditions KKT ($\frac{\partial L}{\partial w} = 0$ and $\frac{\partial L}{\partial \lambda} = 0$) pour arriver au résultat.

Answer 3. 1. BatchNorm prend la moyenne et la variance sur les exemples et les caractéristiques (une valeur par canal) :

$$\boldsymbol{\mu}_{batch} = \begin{bmatrix} 2.5 \\ 2.5 \end{bmatrix} \boldsymbol{\sigma}^2_{batch} = \begin{bmatrix} 1.0833 \\ 1.4167 \end{bmatrix}$$

LayerNorm prend la moyenne et la variance sur les canaux et les caractéristiques (une valeur par exemple) :

$$\boldsymbol{\mu}_{layer} = \left[\left[2.8333\right], \left[2.6667\right], \left[2.5\right], \left[2\right]\right] \boldsymbol{\sigma}^2_{layer} = \left[\left[1.1389\right], \left[1.2222\right], \left[0.9167\right], \left[1.3333\right]\right]$$

InstanceNorm prend la moyenne et la variance sur les caractéristiques (une valeur pour chaque paire canal-exemple) :

$$\boldsymbol{\mu}_{instance} = \left[\begin{bmatrix} 3 \\ 2.6667 \end{bmatrix}, \begin{bmatrix} 2.6667 \\ 2.6667 \end{bmatrix}, \begin{bmatrix} 2.6667 \\ 2.3333 \end{bmatrix}, \begin{bmatrix} 1.6667 \\ 2.3333 \end{bmatrix}\right]$$

$$\boldsymbol{\sigma}^2_{instance} = \left[\begin{bmatrix} 0.6667 \\ 1.5556 \end{bmatrix}, \begin{bmatrix} 1.5556 \\ 0.8889 \end{bmatrix}, \begin{bmatrix} 0.2222 \\ 1.5556 \end{bmatrix}, \begin{bmatrix} 0.8889 \\ 1.5556 \end{bmatrix}\right]$$

2. BatchNorm est pire pour NLP que LayerNorm car il a été observé que les statistiques de lots pour les données dans NLP ont une très grande variance tout au long du training (voir Shen & Yao et al., 2020). Pour une implémentation naïve de BatchNorm, nous serions confrontés au problème de la gestion de différentes constantes de normalisation pour des lots ayant des longueurs de phrases différentes. Cela pourrait conduire à des instabilités pendant l'apprentissage.
3. Soit $\boldsymbol{x}_i$ et $\boldsymbol{r}_i$ les i-èmes rangées de $\boldsymbol{X}$ et $\boldsymbol{R}$. Notez que $\mathbb{E}[\boldsymbol{r}_{ij}] = p$ et $\text{Var}(\boldsymbol{r}_{ij}) = p(1-p)$. Le terme quadratique est une somme d'erreurs quadratiques. En utilisant la relation $\mathbb{E}[Z^2] = \text{Var}(Z) + \mathbb{E}[Z]^2$ et le fait que $\boldsymbol{r}_{ij}$ est i.i.d., nous avons

$$\begin{aligned}
\mathbb{E}[L(\boldsymbol{w})] &= \sum_{i=1}^{n} \mathbb{E}[(\boldsymbol{y}_i - \boldsymbol{w}^\top(\boldsymbol{x}_i \odot \boldsymbol{r}_i))^2] \\
&= \sum_{i=1}^{n} \mathbb{E}[\boldsymbol{y}_i - \boldsymbol{w}^\top(\boldsymbol{x}_i \odot \boldsymbol{r}_i)]^2 + \text{Var}(\boldsymbol{y}_i - \boldsymbol{w}^\top(\boldsymbol{x}_i \odot \boldsymbol{r}_i)) \\
&= \sum_{i=1}^{n} (\boldsymbol{y}_i - p\boldsymbol{w}^\top \boldsymbol{x}_i)^2 + p(1-p) \sum_{j=1}^{d} (\boldsymbol{w}_j \boldsymbol{x}_{ij})^2 \\
&= ||\boldsymbol{y} - p\boldsymbol{X}\boldsymbol{w}||^2 + p(1-p)||\Gamma \boldsymbol{w}||^2
\end{aligned}$$

4. Nous savons $\boldsymbol{w}^{(0)} = 0$. Le gradient par rapport à $\boldsymbol{w}$ est

$$\nabla_{\boldsymbol{w}} f = -\sum_{i=1}^{n} 2(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)\boldsymbol{x}_i$$

Donc, le gradient est une combinaison linéaire de $\boldsymbol{x}_i$s, de la forme

$$\nabla_{\boldsymbol{w}} f = \boldsymbol{X}^\top \boldsymbol{k}$$

où $\boldsymbol{k}$ est un vecteur de valeurs constantes.
En utilisant la règle de la descente de gradient, nous avons

$$\boldsymbol{w}^{(1)} = 0 - \eta \boldsymbol{X}^\top \boldsymbol{k}$$

Ces gradients s'accumulent sur un certain nombre d'itérations et la valeur optimale de $\boldsymbol{w}^{gd}$ trouvée après disons p itérations peut également être représentée comme un vecteur qui est dans le span de $\boldsymbol{x}_i$s. C'est-à-dire que

$$\boldsymbol{w}^{gd} = \boldsymbol{X}^\top \boldsymbol{k}^*$$

Après substitution, nous obtenons

$$\begin{aligned}
\boldsymbol{X}\boldsymbol{w}^{gd} &= \boldsymbol{y} \\
\boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{k}^* &= \boldsymbol{y} \\
\boldsymbol{k}^* &= (\boldsymbol{X}\boldsymbol{X}^\top)^{-1}\boldsymbol{y} \\
\boldsymbol{w}^{gd} &= \boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{X}^\top)^{-1}\boldsymbol{y}
\end{aligned}$$

Le Lagrangien pour le problème d'optimisation donné peut être calculé comme suit

$$L(\boldsymbol{w}, \lambda) = \|\boldsymbol{w}\|^2 + \lambda^\top (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})$$

Soit la solution optimale $\boldsymbol{w}^*$. Les conditions KKT peuvent être calculées comme suit

$$\frac{\partial L}{\partial \boldsymbol{w}^*} = 2\boldsymbol{w}^* + \boldsymbol{X}^\top \lambda = 0$$
$$\frac{\partial L}{\partial \lambda} = \boldsymbol{X}\boldsymbol{w}^* - \boldsymbol{y} = 0$$

En utilisant la première condition, on obtient

$$\boldsymbol{w}^* = -\frac{\boldsymbol{X}^\top \lambda}{2}$$

En substituant la deuxième condition, on obtient

$$\lambda = -2(\boldsymbol{X}\boldsymbol{X}^\top)^{-1}\boldsymbol{y}$$

Ainsi, nous avons

$$\boldsymbol{w}^* = \boldsymbol{X}^\top (\boldsymbol{X}\boldsymbol{X}^\top)^{-1}\boldsymbol{y}$$

Par conséquent, les deux méthodes ci-dessus conduisent à la même solution, donc $\boldsymbol{w}^* = \boldsymbol{w}^{gd}$.
Note: $\boldsymbol{w} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{y}$ n'est pas une solution valide et n'est pas équivalente à la solution ci-dessus car $\boldsymbol{X}^\top \boldsymbol{X}$ n'est pas inversible pour une matrice "grasse" $\boldsymbol{X}$ (c'est-à-dire lorsque $d > n$). Le simple fait de mettre le gradient à 0 pour la solution de descente de gradient pourrait vous conduire à cette solution mais elle n'est pas correcte car (i) l'inverse n'existe pas et (ii) un système sous-déterminé a des solutions infinies, donc la contrainte de norme minimale est implicitement ajoutée par la descente de gradient (c'est pourquoi cette question existe) ! De même, les solutions utilisant $\boldsymbol{X}^{-1}$ sont également incorrectes car l'existence de cet inverse n'est pas garantie.
Voir here et here pour plus d'informations. Le problème d'optimisation consiste à trouver une $\boldsymbol{w}$ qui a une norme minimale et qui satisfait également le système d'équations donné. Nous voyons que la descente de gradient choisit la solution de norme minimale. Comme il s'agit d'un système d'équations sous-déterminé, il existe une infinité de solutions au système d'équations. Les résultats ci-dessus montrent que la descente par gradient a une préférence pour la solution de norme minimale.

Question 4 (20 pts)**.** (**Question de révision d'article**)

dans cette question, vous allez écrire **une page** révision de the vision transformer paper. Veuillez structurer votre examen dans les sections suivantes:

1. **Sommaire [5 pts]**:
   (a) De quoi parle cet article ?
   (b) Quelle est la principale contribution ?

(c) Décrivez l'approche principale et les résultats. Seulement des faits, pas encore d'opinions.

2. **Forces [5 pts]**:
   (a) Y a-t-il un nouvel aperçu théorique ?
   (b) Ou un progrès empirique important ? Ont-ils résolu un problème permanent ?
   (c) Ou une bonne formulation pour un nouveau problème ?
   (d) Des résultats concrets (code, algorithme, etc.) ?
   (e) Les expériences sont-elles bien exécutées ?
   (f) Utile pour la communauté en général ?
3. **Faiblesses [5 pts]** :
   (a) Que peut-on faire de mieux ?
   (b) Des bases de référence manquantes ? Des ensembles de données manquants ?
   (c) Des choix de conception bizarres dans l'algorithme ne sont-ils pas bien expliqués ? Qualité de l'écriture ?
   (d) Est-ce qu'il y a suffisamment de nouveauté dans ce qu'ils proposent ? Variation mineure de travaux antérieurs ?
   (e) Pourquoi devrait-on s'en soucier ? Le problème est-il intéressant et important ?
4. **Reflets [5 pts]**:
   (a) Quel est le lien avec d'autres concepts que vous avez vus dans la classe ?
   (b) Quelles sont les prochaines orientations de recherche dans ce domaine ?
   (c) Quelles nouvelles idées (directement ou indirectement liées) ce document vous a-t-il donné ? Qu'est-ce que tu voudrais essayer ?

Answer 4. Cette question est subjective, nous accepterons donc une variété de réponses. On attend de vous que vous analysiez le document et que vous proposiez votre propre point de vue et vos idées (plus que ce que le document propose).