**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 $\sigma$ 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.

   **Réponse**

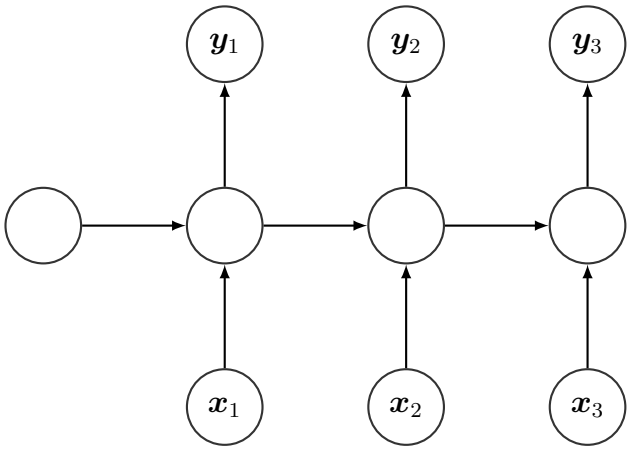


FIGURE 1 – Graphique de calcul de l'unité récurrente déroulé en trois temps.

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.** 1. Le Graphique complété :

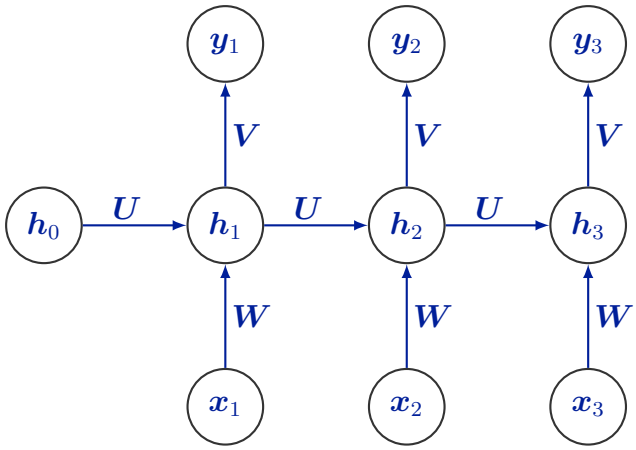


FIGURE 2 – Graphique de calcul de l'unité récurrente déroulé en trois temps.

2. On a :

$$L = \sum_t L_t$$

$$\iff \nabla_{\boldsymbol{h}_t} L = \frac{\partial}{\partial \boldsymbol{h}_t} \sum_k L_k$$

Or, pour un $t$ donné, seulement les termes $L_k$ avec $k \geq t$ dépendent de $h_t$. On peut donc poser :

$$\begin{aligned}
\nabla_{\boldsymbol{h}_t} L &= \frac{\partial}{\partial \boldsymbol{h}_t} \sum_{k \geq t} L_k \\
&= \frac{\partial}{\partial \boldsymbol{h}_t} L_t + \frac{\partial}{\partial \boldsymbol{h}_t} \sum_{k \geq t+1} L_k \\
&= \nabla_{\boldsymbol{h}_t} L_t + (\frac{\partial \boldsymbol{h}_{t+1}}{\partial \boldsymbol{h}_t})^T (\frac{\partial}{\partial \boldsymbol{h}_{t+1}}) \sum_{k \geq t+1} L_k
\end{aligned}$$

Or, de façon analogue avec $t+1$, pour un $t+1$ donné, seulement les termes $L_k$ avec $k \geq t+1$ dépendent de $h_{t+1}$. On a alors : $(\frac{\partial}{\partial \boldsymbol{h}_{t+1}}) \sum_{k \geq t+1} L_k = \frac{\partial L}{\partial \boldsymbol{h}_{t+1}}$
D'où :

$$\begin{aligned}
\nabla_{\boldsymbol{h}_t} L &= \nabla_{\boldsymbol{h}_t} L_t + (\frac{\partial \boldsymbol{h}_{t+1}}{\partial \boldsymbol{h}_t})^T (\frac{\partial}{\partial \boldsymbol{h}_{t+1}}) \sum_{k \geq t+1} L_k \\
&= \nabla_{\boldsymbol{h}_t} L_t + (\frac{\partial \boldsymbol{h}_{t+1}}{\partial \boldsymbol{h}_t})^T \frac{\partial L}{\partial \boldsymbol{h}_{t+1}} \\
&= \nabla_{\boldsymbol{h}_t} L_t + (\frac{\partial \boldsymbol{h}_{t+1}}{\partial \boldsymbol{h}_t})^T \nabla_{\boldsymbol{h}_{t+1}} L
\end{aligned}$$

Donc :

$$\boxed{\nabla_{\boldsymbol{h}_t} L = \nabla_{\boldsymbol{h}_t} L_t + (\frac{\partial \boldsymbol{h}_{t+1}}{\partial \boldsymbol{h}_t})^T \quad \nabla_{\boldsymbol{h}_{t+1}} L}$$

3. Calcul de $\nabla_{\boldsymbol{h}_t} L_t$ et $\frac{\partial \boldsymbol{h}_{t+1}}{\partial \boldsymbol{h}_t}$ :
Calcul de $\nabla_{\boldsymbol{h}_t} L_t$:

$$\begin{aligned}
\nabla_{\boldsymbol{h}_t} L_t &= \frac{\partial}{\partial \boldsymbol{h}_t} L_t \\
&= \frac{\partial}{\partial \boldsymbol{h}_t} ||\boldsymbol{z}_t - \boldsymbol{y}_t||_2^2 \\
&= (\frac{\partial \boldsymbol{y}_t}{\partial \boldsymbol{h}_t})^T (\frac{\partial}{\partial \boldsymbol{y}_t}) ||\boldsymbol{z}_t - \boldsymbol{y}_t||_2^2 \\
&= (\frac{\partial}{\partial \boldsymbol{h}_t} (\boldsymbol{V}\boldsymbol{h}_t))^T (\frac{\partial}{\partial \boldsymbol{y}_t}) (\boldsymbol{y}_t^T \boldsymbol{y}_t - 2\boldsymbol{y}_t \boldsymbol{z}_t + \boldsymbol{z}_t^T \boldsymbol{z}_t) \\
&= \boldsymbol{V}^T (2\boldsymbol{y}_t - \boldsymbol{z}_t) \\
&= -2(\boldsymbol{z}_t - \boldsymbol{y}_t) V
\end{aligned}$$

Calcul de $\frac{\partial \boldsymbol{h}_{t+1}}{\partial \boldsymbol{h}_t}$, avec : $\boldsymbol{h}_{t+1} = \sigma(\boldsymbol{W}\boldsymbol{x}_{t+1} + \boldsymbol{U}\boldsymbol{h}_t)$ :

$$\frac{\partial \boldsymbol{h}_{t+1}}{\partial \boldsymbol{h}_t} = diag(\sigma'(\boldsymbol{W}\boldsymbol{x}_{t+1} + \boldsymbol{U}\boldsymbol{h}_t)) U$$

Or, on sait que :

$$\sigma'(X) = \sigma(X)(1 - \sigma(X))$$

avec

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

Donc :

$$\sigma'(\boldsymbol{W}\boldsymbol{x}_{t+1} + \boldsymbol{U}\boldsymbol{h}_t) = \boldsymbol{h}_{t+1}(1 - \boldsymbol{h}_{t+1})$$

et

$$\frac{\partial \boldsymbol{h}_{t+1}}{\partial \boldsymbol{h}_t} = diag(\boldsymbol{h}_{t+1}(1 - \boldsymbol{h}_{t+1})) U$$

Conclusion :

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

et

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

On a alors :

$$\boxed{\nabla_{\boldsymbol{h}_t} L = \nabla_{\boldsymbol{h}_t} L_t = -2(\boldsymbol{z}_t - \boldsymbol{y}_t)V + (diag(\boldsymbol{h}_{t+1}(1 - \boldsymbol{h}_{t+1}))U)^T \ \ \nabla_{\boldsymbol{h}_{t+1}} L}$$

4. Calcul de $\nabla_{\boldsymbol{W}} L$ et $\nabla_{\boldsymbol{U}} L$ en fonction de $\nabla_{\boldsymbol{h}_t} L$.
Pour se faire, on prendra en compte la contribution des matrices à un moment particulier $t$ tel que :

$$\nabla_{\boldsymbol{W}} L = \sum_t \nabla_{\boldsymbol{W}_t} L$$
$$\nabla_{\boldsymbol{U}} L = \sum_t \nabla_{\boldsymbol{U}_t} L$$

Calcul de $\nabla_{\boldsymbol{W}_t} L$ :

$$\begin{aligned}\nabla_{\boldsymbol{W}_t} L &= \frac{\partial L}{\partial \boldsymbol{W}_t} \\ &= \frac{\partial L}{\partial \boldsymbol{h}_t} \cdot \frac{\partial \boldsymbol{h}_t}{\partial \boldsymbol{W}_t} \\ &= diag(\boldsymbol{h}_t(1 - \boldsymbol{h}_t))(\nabla_{\boldsymbol{h}_t} L)\boldsymbol{x}_t^T\end{aligned}$$

Donc :

$$\boxed{\nabla_{\boldsymbol{W}} L = \sum_t diag(\boldsymbol{h}_t(1 - \boldsymbol{h}_t))(\nabla_{\boldsymbol{h}_t} L)\boldsymbol{x}_t^T}$$

Calcul de $\nabla_{\boldsymbol{U}_t} L$ :

$$\begin{aligned}\nabla_{\boldsymbol{U}_t} L &= \frac{\partial L}{\partial \boldsymbol{U}_t} \\ &= \frac{\partial L}{\partial \boldsymbol{h}_t} \cdot \frac{\partial \boldsymbol{h}_t}{\partial \boldsymbol{U}_t} \\ &= diag(\boldsymbol{h}_t(1 - \boldsymbol{h}_t))(\nabla_{\boldsymbol{h}_t} L)\boldsymbol{h}_{t-1}^T\end{aligned}$$

Donc :

$$\boxed{\nabla_{\boldsymbol{U}} L = \sum_t diag(\boldsymbol{h}_t(1 - \boldsymbol{h}_t))(\nabla_{\boldsymbol{h}_t} L)\boldsymbol{h}_{t-1}^T}$$

5. Les estimations du gradient du BPTT tronqué sont biaisés, du fait qu'elles ne bénéficient pas des garanties de convergence de la théroei du gradient stochastique.
Même si le BPTT tronqué est un moyen pour atténuer l'explosion ou la disparition des gradients, il peut parfois poser certains problèmes lorsqu'on considère des tâches impliquant la langue :
En effet, la troncature favorise les dépendances à court terme et pose ainsi problème lorsqu'on considère des données de longues séquences.

**Question 2** (5-5-3-5-2)**.** (**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

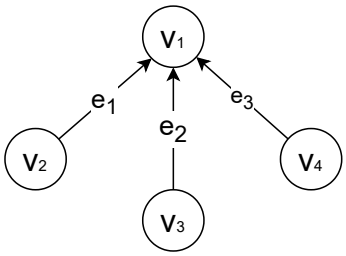


FIGURE 3 – Graphique G pour Q2.2

:

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

*(Ce sont des vecteurs de rangée.)*
la fonction d'agrégation $Agg(H'_{1t})$ is:

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

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 $h_1^{t+1}$ ?
*($h_i^t$ et $W$ sont des vecteurs de rangée.)*

(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. On a :

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

et

$$Agg(H'_{it}) = \sum_j f_{ji}(h_j^t) \quad | \quad \forall j, v_j \in \mathcal{N}(v_i)$$

On a donc :

$$\boxed{h_i^{t+1} = q(h_i^t, \sum_{j, v_j \in \mathcal{N}(v_i)} f_{ji}(h_j^t))}$$

2. On a :

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

Or :

$$W(h_1^t)^T = [1, 1][1, -1]^T = 0$$

et

$$\begin{aligned} Agg(H'_{1t}) &= [0.6, 0.2, 0.2][[-2, 2][0, -2][2, 0]] \\ &= [-0.8, 0.8] \end{aligned}$$

Donc :

$$\boxed{h_1^{t+1} = 0 + [0, 0.8] = [0, 0.8]}$$

3. Graphe entièrement connecté " I eat red apples".

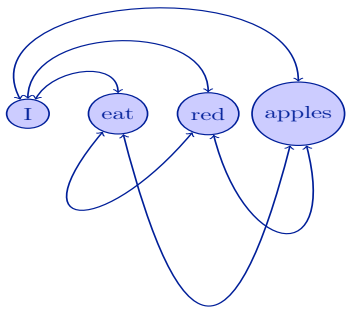


FIGURE 4 – Graphie G : "I eat red apples"

Le graphique G a 12 arrêtes : $n \times (n-1)$, avec n le nombre de sommets du graphe.

4. L'équation standard pour la mise à jour de l'attention à la couche $l$ **pour le Transformer** s'écrit:

$$h_i^{l+1} = \sum_{j \in S} \left(\texttt{softmax}_j(Q^l h_i^l \cdot K^l h_j^l) V^l h_j^l\right)$$

De plus, d'après la question 1 :

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

Ainsi, pour que le mécanisme d'attention à tête unique du Transformer soit équivalent au GNN, il faut tout d'abord définir la fonction de mise à jour $q$ et la fonction $f$ tel que :

$$q(h_i^t, \sum_{j, v_j \in \mathcal{N}(v_i)} f_{ji}(h_j^t)) = \sum_{j, v_j \in \mathcal{N}(v_i)} f_{ji}(h_j^t)$$

et

$$f_{ji}(h_j^t)) = \left(\texttt{softmax}_j(Q^l h_i^l \cdot K^l h_j^l) V^l h_j^l\right) \quad | \quad \forall j, v_j \in \mathcal{N}(v_i)$$

Avec les fonctions $q$ et $f$ définies de la sorte, on a alors :

$$\begin{aligned} h_i^{t+1} = q(h_i^t, \sum_{j, v_j \in \mathcal{N}(v_i)} f_{ji}(h_j^t)) &= \sum_{j, v_j \in \mathcal{N}(v_i)} f_{ji}(h_j^t) \\ &= \sum_{j, v_j \in \mathcal{N}(v_i)} \left(\texttt{softmax}_j(Q^l h_i^l \cdot K^l h_j^l) V^l h_j^l\right) \end{aligned}$$

Il faut maintenant que :

$$\sum_{j, v_j \in \mathcal{N}(v_i)} \left(\texttt{softmax}_j(Q^l h_i^l \cdot K^l h_j^l) V^l h_j^l\right) = \sum_{j \in S} \left(\texttt{softmax}_j(Q^l h_i^l \cdot K^l h_j^l) V^l h_j^l\right)$$

Pour se faire, il suffit seulement que le GNN ait en entrée un graphe **complètement connecté**. En effet, le Transformer calcule the pair-wise attention entre tous les noeuds/positions de l'entrée. Il faut donc supposer que, dans le cas du GNN, le graphe est complètement connecté.
Dans ce cas précis on aura bien :

$$\begin{aligned} h_i^{t+1} = q(h_i^t, \sum_{j, v_j \in \mathcal{N}(v_i)} f_{ji}(h_j^t)) &= \sum_{j, v_j \in \mathcal{N}(v_i)} \left(\texttt{softmax}_j(Q^l h_i^l \cdot K^l h_j^l) V^l h_j^l\right) \\ &= \sum_{j \in S} \left(\texttt{softmax}_j(Q^l h_i^l \cdot K^l h_j^l) V^l h_j^l\right) \end{aligned}$$

Ainsi, le single head attention mecanism du Transformer est équivalent au GNN si celui-ci reçoit un graphe entièrement connecté (fully connected graph) en entrée, et qu'on considère la fonction de mise à jour $q$ et la fonction $f$ définies précedemment.

5. Un problème récurrent avec les GNN est qu'ils rendent difficile les dépendances à très long terme entre les mots.
En effet, le nombre d'arrêtes du graphe évolue de manière quadratique avec le nombre de noeuds. Par exemple, dans une phrase à $n$ noeuds, un Transformer / GNN effectueraient des calculs sur $n^2$ pairs de mots. Lorsque $n$ est très grand, ce nombre peut rapidement exploser.
La complexité en temps est quadratique : $O(n^2)$.

**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 $\Gamma$ 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 :

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

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 $\eta$ 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:

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

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

*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 $\lambda^*$ 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. On considère ici le tenseur $\boldsymbol{X} \in \mathbb{R}^{T\times C\times W}$ avec $T$ le nombre de batches (ici 4), $C$ le nombre de canaux (channels, ici 2) et $W$ le nombre de caractéristiques (features, ici 3).

**Batch Normalization**

Par définition du mean batch normalization :

$$\boldsymbol{\mu}_i = \frac{1}{TW}\sum_{t=1}^{T}\sum_{w=1}^{W}\boldsymbol{X}_{tiw} = \frac{1}{12}\sum_{t=1}^{4}\sum_{w=1}^{3}\boldsymbol{X}_{tiw}$$

D'où :

$$\boxed{\boldsymbol{\mu}_{batch} = [\frac{5}{2}, \frac{5}{2}] = [2.5, 2.5]}$$

Par définition de la batch variance :

$$\boldsymbol{\sigma}_i^2 = \frac{1}{TW}\sum_{t=1}^{T}\sum_{w=1}^{W}(\boldsymbol{X}_{tiw} - \boldsymbol{\mu}_i)^2 = \frac{1}{12}\sum_{t=1}^{4}\sum_{w=1}^{3}(\boldsymbol{X}_{tiw} - \boldsymbol{\mu}_i)^2$$

D'où :

$$\boxed{\boldsymbol{\sigma}^2_{batch} = [1.083, 1.416]}$$

**Layer Normalization**

Par définition du mean layer normalization :

$$\boldsymbol{\mu}_t = \frac{1}{CW}\sum_{i=1}^{C}\sum_{w=1}^{W}\boldsymbol{X}_{tiw} = \frac{1}{6}\sum_{i=1}^{2}\sum_{w=1}^{3}\boldsymbol{X}_{tiw}$$

D'où :

$$\boxed{\boldsymbol{\mu}_{layer} = [\frac{17}{6}, \frac{8}{3}, \frac{5}{2}, 2] = [2.83, 2.67, 2.5, 2]}$$

Par définition de la layer variance :

$$\boldsymbol{\sigma}^2_t = \frac{1}{CW}\sum_{i=1}^{C}\sum_{w=1}^{W}(\boldsymbol{X}_{tiw} - \boldsymbol{\mu}_t)^2 = \frac{1}{12}\sum_{i=1}^{4}\sum_{w=1}^{3}(\boldsymbol{X}_{tiw} - \boldsymbol{\mu}_t)^2$$

D'où :

$$\boxed{\boldsymbol{\sigma}^2_{layer} = [\frac{41}{36}, \frac{11}{9}, 0.916, \frac{4}{3}] = [1.139, 1.22, 0.916, 1.33]}$$

**Instance Normalization**

Par définition du mean instance normalization :

$$\boldsymbol{\mu}_{ti} = \frac{1}{W}\sum_{w=1}^{W}\boldsymbol{X}_{tiw} = \frac{1}{3}\sum_{w=1}^{3}\boldsymbol{X}_{tiw}$$

D'où :

$$\boxed{\boldsymbol{\mu}_{instance} = [[3, 2.67], [2.67, 2.67], [2.67, 2.33], [1.67, 2.33]]}$$

Par définition de la instance variance :

$$\boldsymbol{\sigma}^2_{ti} = \frac{1}{W}\sum_{w=1}^{W}(\boldsymbol{X}_{tiw} - \boldsymbol{\mu}_{ti})^2 = \frac{1}{3}\sum_{w=1}^{3}(\boldsymbol{X}_{tiw} - \boldsymbol{\mu}_{ti})^2$$

D'où :

$$\boxed{\boldsymbol{\sigma}^2_{instance} = [[0.66, 1.56], [1.56, 0.89], [0.22, 1.56], [0.89, 1.56]]}$$

2. Il existe des différences dans les batch statistics des données NLP comparés à celles de Computer Vision. En effet, dans le cas des NLP, on observe une très grande variance tout au long de l'entraînement. On retrouve cette variance même dans les gradients correspondants. Ce qui n'est pas de le cas des données de Computer Vision où on observe des données de faible variance.
Les statistiques des données NLP après avoir effectué la Batch Normalization présentent de grandes fluctuations tout au long de l'entraînement, ce qui entraîne une instabilité.

3. On considère un modèle de régression linéaire avec l'entrée $\boldsymbol{X} \in \mathbb{R}^{n\times d}$, poids $\boldsymbol{w} \in \mathbb{R}^{d\times 1}$ et objectifs $\boldsymbol{y} \in \mathbb{R}^{n\times 1}$. De plus, on a $\boldsymbol{R} \in \mathbb{R}^{n\times d}$ le masque dropout tel que $\boldsymbol{R}_{ij} \sim \text{Bern}(p)$ est **échantillonné i.i.d.** de la distribution Bernoulli.
On a :

$$L(\boldsymbol{w}) = ||\boldsymbol{y} - (\boldsymbol{X} \odot \boldsymbol{R})\boldsymbol{w}||^2$$
$$L(\boldsymbol{w}) = \sum_i \sum_j (\boldsymbol{y}_i - (\boldsymbol{X}_{ij}\boldsymbol{R}_{ij})\boldsymbol{w}_j)^2$$

On cherche à calculer :

$$\begin{aligned}\mathbb{E}[L(\boldsymbol{w})] &= \mathbb{E}[\sum_i \sum_j (\boldsymbol{y}_i - (\boldsymbol{X}_{ij}\boldsymbol{R}_{ij})\boldsymbol{w}_j)^2] \\ &= \sum_i \sum_j \mathbb{E}[(\boldsymbol{y}_i - (\boldsymbol{X}_{ij}\boldsymbol{R}_{ij})\boldsymbol{w}_j)^2]\end{aligned}$$

En utilisant : $\text{Var}(Z) = \mathbb{E}[Z^2] - \mathbb{E}[Z]^2$, on a $\mathbb{E}[Z^2] = \mathbb{E}[Z]^2 + \text{Var}(Z)$.
On considérant $Z = (\boldsymbol{y}_i - (\boldsymbol{X}_{ij}\boldsymbol{R}_{ij})\boldsymbol{w}_j)^2$, on a alors :

$$\begin{aligned}\mathbb{E}[L(\boldsymbol{w})] &= \sum_i \sum_j \mathbb{E}[(\boldsymbol{y}_i - (\boldsymbol{X}_{ij}\boldsymbol{R}_{ij})\boldsymbol{w}_j)^2] \\ &= \sum_i \sum_j (\mathbb{E}[\boldsymbol{y}_i - (\boldsymbol{X}_{ij}\boldsymbol{R}_{ij})\boldsymbol{w}_j]^2 + \text{Var}(\boldsymbol{y}_i - (\boldsymbol{X}_{ij}\boldsymbol{R}_{ij})\boldsymbol{w}_j)) \\ &= \sum_i \sum_j \mathbb{E}[\boldsymbol{y}_i - (\boldsymbol{X}_{ij}\boldsymbol{R}_{ij})\boldsymbol{w}_j]^2 + \sum_i \sum_j \text{Var}(\boldsymbol{y}_i - (\boldsymbol{X}_{ij}\boldsymbol{R}_{ij})\boldsymbol{w}_j)\end{aligned}$$

Montrons que : $\sum_i \sum_j \mathbb{E}[\boldsymbol{y}_i - (\boldsymbol{X}_{ij}\boldsymbol{R}_{ij})\boldsymbol{w}_j]^2 = ||\boldsymbol{y} - p\boldsymbol{X}\boldsymbol{w}||^2$.
Sachant que : $\mathbb{E}[aX + b] = a\mathbb{E}[X] + b$, et que $\boldsymbol{R}_{ij} \sim \text{Bern}(p)$ est **échantillonné i.i.d.**, on a :

$$\sum_i \sum_j \mathbb{E}[\boldsymbol{y}_i - (\boldsymbol{X}_{ij}\boldsymbol{R}_{ij})\boldsymbol{w}_j]^2 = \sum_i \sum_j (\boldsymbol{y}_i - \mathbb{E}[\boldsymbol{R}_{ij}]\boldsymbol{X}_{ij}\boldsymbol{w}_j)^2$$

De plus, $\boldsymbol{R}_{ij} \sim \text{Bern}(p)$, on a alors $\mathbb{E}[\boldsymbol{R}_{ij}] = p$. D'où :

$$\begin{aligned}\sum_i \sum_j \mathbb{E}[\boldsymbol{y}_i - (\boldsymbol{X}_{ij}\boldsymbol{R}_{ij})\boldsymbol{w}_j]^2 &= \sum_i \sum_j (\boldsymbol{y}_i - \mathbb{E}[\boldsymbol{R}_{ij}]\boldsymbol{X}_{ij}\boldsymbol{w}_j)^2 \\ &= \sum_i \sum_j (\boldsymbol{y}_i - p\boldsymbol{X}_{ij}\boldsymbol{w}_j)^2 \\ &= ||\boldsymbol{y} - p\boldsymbol{X}\boldsymbol{w}||^2\end{aligned}$$

Montrons maintenant que : $\sum_i \sum_j \text{Var}(\boldsymbol{y}_i - (\boldsymbol{X}_{ij}\boldsymbol{R}_{ij})\boldsymbol{w}_j) = p(1-p)||\Gamma\boldsymbol{w}||^2$.
Sachant que : $\text{Var}[aX + b] = a^2\text{Var}[X]$, et que $\boldsymbol{R}_{ij} \sim \text{Bern}(p)$ est **échantillonné i.i.d.**, on a :

$$\begin{aligned}\sum_i \sum_j \text{Var}(\boldsymbol{y}_i - (\boldsymbol{X}_{ij}\boldsymbol{R}_{ij})\boldsymbol{w}_j) &= \sum_i \sum_j \text{Var}(\boldsymbol{X}_{ij}\boldsymbol{R}_{ij}\boldsymbol{w}_j) \\ &= \sum_i \sum_j (\boldsymbol{X}_{ij}\boldsymbol{w}_j)^2 \text{Var}(\boldsymbol{R}_{ij})\end{aligned}$$

De plus, $\boldsymbol{R}_{ij} \sim \text{Bern}(p)$, on a alors $\text{Var}(\boldsymbol{R}_{ij}) = p(1-p)$.
D'où :

$$\begin{aligned}\sum_i \sum_j \text{Var}(\boldsymbol{y}_i - (\boldsymbol{X}_{ij}\boldsymbol{R}_{ij})\boldsymbol{w}_j) &= \sum_i \sum_j (\boldsymbol{X}_{ij}\boldsymbol{w}_j)^2 \text{Var}(\boldsymbol{R}_{ij}) \\ &= \sum_i \sum_j (\boldsymbol{X}_{ij}\boldsymbol{w}_j)^2 p(1-p) \\ &= p(1-p) \sum_i \sum_j \boldsymbol{w}_j^T \boldsymbol{X}_{ij}^T \boldsymbol{X}_{ij} \boldsymbol{w}_j \\ &= p(1-p) \sum_i \boldsymbol{w}^T \boldsymbol{X}_i^T \boldsymbol{X}_i \boldsymbol{w}\end{aligned}$$

Or, par définition, $\Gamma_{ii} = (\boldsymbol{X}^\top \boldsymbol{X})_{ii}^{1/2}$. On a alors :

$$\begin{aligned}\sum_i \sum_j \text{Var}(\boldsymbol{y}_i - (\boldsymbol{X}_{ij}\boldsymbol{R}_{ij})\boldsymbol{w}_j) &= p(1-p) \sum_i \boldsymbol{w}^T \boldsymbol{X}_i^T \boldsymbol{X}_i \boldsymbol{w} \\ &= p(1-p) \sum_i \boldsymbol{w}^T \Gamma_{ii}^2 \boldsymbol{w} \\ &= p(1-p) ||\Gamma \boldsymbol{w}||^2\end{aligned}$$

On a alors :

$$\boxed{\mathbb{E}[L(\boldsymbol{w})] = \sum_i \sum_j \mathbb{E}[\boldsymbol{y}_i - (\boldsymbol{X}_{ij}\boldsymbol{R}_{ij})\boldsymbol{w}_j]^2 + \sum_i \sum_j \text{Var}(\boldsymbol{y}_i - (\boldsymbol{X}_{ij}\boldsymbol{R}_{ij})\boldsymbol{w}_j)}$$

avec

$$\boxed{\sum_i \sum_j \mathbb{E}[\boldsymbol{y}_i - (\boldsymbol{X}_{ij}\boldsymbol{R}_{ij})\boldsymbol{w}_j]^2 = ||\boldsymbol{y} - p\boldsymbol{X}\boldsymbol{w}||^2}$$

et

$$\boxed{\sum_i \sum_j \text{Var}(\boldsymbol{y}_i - (\boldsymbol{X}_{ij}\boldsymbol{R}_{ij})\boldsymbol{w}_j) = p(1-p)||\Gamma \boldsymbol{w}||^2}$$

Conclusion :

$$\boxed{\mathbb{E}[L(\boldsymbol{w})] = ||\boldsymbol{y} - p\boldsymbol{X}\boldsymbol{w}||^2 + p(1-p)||\Gamma \boldsymbol{w}||^2}$$

4.

**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.** One page review

1. **Résumé** : L'article parle de l'application des Transformers à des tâches de reconaissance d'images. En effet, il est dit que les CNN ne sont pas indispensables et qu'un transformateur peut très bien fonctionner, voir "remplacer" les meilleurs CNN. La principale contribution de cet article est l'application d'un Transformer standard directement aux images, avec le moins de modifications possibles, pour résoudre une tâche de classification d'images supervisée. L'approche expliquée est la suivante : l'image est divisée en patchs et la séquence d'incorporations linéaires de ces patchs est donnée en entrée d'un Transformer. Les patchs d'image sont traités de la même manière que les tokens (mots) lorsqu'on exécute une tâche de NLP.
   ViT atteint d'excellents résultats par rapport aux réseaux convolutifs tout en nécessitant beaucoup moins de ressources de calcul pour l'entraînement. En effet, le meilleur modèle atteint la précision de 88,55 % sur ImageNet, 90,72 % sur ImageNet-ReaL, 94,55 % sur CIFAR-100 et 77,63 % sur la suite VTAB (suite of 19 tasks).

2. **Forces** : Selon moi, il y a un nouvel aperçu théorique : en raison du coût du calcul quadratique du mécanisme d'attention, le modèle Transformer ne pouvait pas être utilisé naïvement pour les images. Selon moi, il y a un progrès notable notamment lorsqu'on observe que le Vision Transformer (ViT) présente de meilleurs résultats que les CNN à la pointe de la technologie, tout en nécessitant beaucoup moins de ressources de calcul pour s'entraîner. Le problème qu'ils ont résolu est le fait d'enfin pouvoir utiliser le Transformer pour des tâches de classification d'images, et ce, en obtenant des résultats extrêmement convaincants. Un nouveau problème qui émmerge est le fait de devoir entraîner le modèle sur un important volume de données (300M d'images labelisées).

   Plus généralement, le papier est très bien écrit, les travaux de recherche sont tous clairement énoncés et résumés. Les expériences semblent être bien exécutées, l'algorithme utilisé est bien retranscrit et expliqué. L'évaluation est approfondie et l'analyse est bonne. Cet article démontre la puissance du modèle Vision Transformer par de vastes expériences à grande échelle, surpassant les meilleurs modèles CNN. Les évaluations comparatives avec des références importantes, des modèles ResNet et hybrides, sont bien conçues et menées avec différentes échelles d'ensembles de données et de modèles. Les résultats sont intéressants pour la communeauté. Les discussions supplémentaires en annexe sont également utiles pour comprendre le modèle.

3. **Faiblesses** : En réalité, le papier ne présente pas de nouveauté technique significative.En effet, le modèle proposé consiste en des modifications, adaptations, du transformer d'origine et de ses variantes existantes. De plus, pour la communeauté en générale, si on ne dispose pas d'un volume énorme de données d'entraînement, il est préférable de continuer d'utiliser des CNN strandards plutôt qu'un modèle Vision Transformer (ViT), à moins d'utiliser un modèle pré-entraîné. Enfin, il y a, à mon avis, un manque d'analyse plus approfondie concernant le biais inductif. Il est dit page 7 : "the convolutional inductive bias is useful for smaller datasets, but for larger ones, learning the relevant patterns directly from data is sufficient, even beneficial." Cependant, on ne sait pas quel biais inductif de CNN empêche une meilleure généralisation : il existe différents biais inductifs dans la convolution standard, il se peut que certains d'entre eux aident à la généralisation et que d'autres non.

4. **Reflexion** : Il existe un lien fort entre le Vision Transformer (ViT) et le Transformer standard vu en classe. En effet, l'architecture du Transformer d'origine est conservée (avec notamment le Transformer Encoder). De plus, on retrouve des notions d'optimization, regularization et normalization, en plus des notions de convolution vues en cours.
   De futurs orientations de recherche seraient de mieux exploiter les architectures hybrides. En effet, il serait interessant à mon avis de mieux approfondir les recherches avec les modèles hybrides, qui semblent présenter des résultats convaincants et très encourageants.