Noms:

- **Shayan Nicolas Hollet (Matricule: 20146766)**
- **Byungsuk Min (Matricule: 20234231)**

Question 1

[5 pts] Quelle est la dérivée du terme de régularisation de la fonction de perte

$$\frac{C}{2}\sum_{j'=1}^{m}\left\|\mathbf{w}^{j'}\right\|_2^2$$

par rapport à w_k^j ? (le $k^{i\,\text{ème}}$ poids du vecteur de poids pour la $j^{i\,\text{ème}}$ classe)? Écrivez tous les étapes et mettez la réponse dans votre fichier PDF.

Solution

Dans le cadre de l'implémentation d'un SVM un-contre-tous avec pénalité L2 pour la classification multi-classe, il est essentiel de calculer les dérivées nécessaires pour la mise à jour des paramètres lors de l'optimisation par descente de gradient. En particulier, nous devons déterminer la dérivée du terme de régularisation L2 de la fonction de perte par rapport à chaque poids individuel.

La fonction de régularisation est donnée par :

$$R(\mathbf{w}) = \frac{C}{2}\sum_{j'=1}^{m}\left\|\mathbf{w}^{j'}\right\|_2^2$$

où :

- C est le paramètre de régularisation.
- m est le nombre de classes.
- $\mathbf{w}^{j'} \in \mathbb{R}^p$ est le vecteur de poids pour la classe j'.
- p est le nombre de caractéristiques (dimensions).

Dérivation de la Dérivée par rapport à w_k^j

Nous cherchons à calculer :

$$\frac{\partial R}{\partial w_k^j}$$

où w_k^j est le k-ième composant du vecteur de poids $\mathbf{w}^j$ correspondant à la classe j.

Étape 1 : Développer le Terme de Régularisation

Le terme de régularisation peut être développé en explicitant la norme L2 :

$$R(\mathbf{w}) = \frac{C}{2}\sum_{j'=1}^{m}\left(\mathbf{w}^{j'}\cdot\mathbf{w}^{j'}\right) = \frac{C}{2}\sum_{j'=1}^{m}\sum_{l=1}^{p}\left(w_l^{j'}\right)^2$$

Étape 2 : Calculer la Dérivée Partielle

La dérivée partielle de $R(\mathbf{w})$ par rapport à w_k^j s'écrit :

$$\frac{\partial R}{\partial w_k^j} = \frac{\partial}{\partial w_k^j} \left(\frac{C}{2} \sum_{j'=1}^{m} \sum_{l=1}^{p} \left(w_l^{j'} \right)^2 \right)$$

Étape 3 : Extraire les Constantes

Les constantes peuvent être sorties de la dérivée :

$$\frac{\partial R}{\partial w_k^j} = \frac{C}{2} \cdot \frac{\partial}{\partial w_k^j} \left(\sum_{j'=1}^{m} \sum_{l=1}^{p} \left(w_l^{j'} \right)^2 \right)$$

Étape 4 : Appliquer la Dérivée aux Sommes

La dérivée d'une somme est la somme des dérivées individuelles :

$$\frac{\partial R}{\partial w_k^j} = \frac{C}{2} \sum_{j'=1}^{m} \sum_{l=1}^{p} \frac{\partial}{\partial w_k^j} \left(\left(w_l^{j'} \right)^2 \right)$$

Étape 5 : Calculer la Dérivée des Termes Individuels

La dérivée de $\left(w_l^{j'} \right)^2$ par rapport à w_k^j est :

$$\frac{\partial}{\partial w_k^j} \left(\left(w_l^{j'} \right)^2 \right) = 2 w_l^{j'} \cdot \frac{\partial w_l^{j'}}{\partial w_k^j}$$

Sachant que :

$$\frac{\partial w_l^{j'}}{\partial w_k^j} = \delta_{ll} \delta_{j'j} = \delta_{lk} \delta_{j'j}$$

où δ_{ab} est le symbole de Kronecker :

$$\delta_{ab} = \begin{cases} 1 & \text{si } a = b \\ 0 & \text{si } a \neq b \end{cases}$$

Donc :

$$\frac{\partial}{\partial w_k^j} \left(\left(w_l^{j'} \right)^2 \right) = 2 w_l^{j'} \cdot \delta_{lk} \delta_{j'j}$$

Étape 6 : Simplifier la Somme

En remplaçant dans la somme, nous obtenons :

$$\frac{\partial R}{\partial w_k^j} = \frac{C}{2} \sum_{j'=1}^{m} \sum_{l=1}^{p} 2 w_l^{j'} \cdot \delta_{lk} \delta_{j'j}$$

Simplifions les sommes en utilisant les propriétés du symbole de Kronecker :

- La somme sur l ne donne un terme non nul que lorsque $l = k$.
- La somme sur j' ne donne un terme non nul que lorsque $j' = j$.

Ainsi, la somme se réduit à :

$$\frac{\partial R}{\partial w_k^j} = \frac{C}{2} \cdot 2 w_k^j = C w_k^j$$

Réponse Finale

La dérivée du terme de régularisation par rapport à w_k^j est donc :

$$\frac{\partial R}{\partial w_k^j} = C w_k^j$$

Conclusion

Ce résultat indique que le gradient du terme de régularisation L2 par rapport à un poids individuel w_k^j est simplement proportionnel à ce poids lui-même, avec le facteur de proportionnalité étant le paramètre de régularisation C. Ceci est cohérent avec l'interprétation de la régularisation L2, qui tend à pénaliser les grands poids pour éviter le surapprentissage et favoriser la généralisation du modèle.

En pratique, lors de l'optimisation de la fonction de perte totale par descente de gradient, ce terme de dérivée est ajouté au gradient calculé à partir des données, ce qui ajuste les poids en conséquence.

Question 2

[10 pts] Quelle est la dérivée du terme appelé hinge loss

$$\frac{1}{n} \sum_{(\mathbf{x}_i, y_i) \in S} \sum_{j'=1}^{m} \mathcal{L}\left(\mathbf{w}^{j'}; (\mathbf{x}_i, y_i)\right)$$

de la fonction de perte par rapport à w_k^j ?
Exprimez votre réponse en termes de $\mathbf{x}_{i,k}$ (la $k^{ième}$ entrée du $i^{ième}$ exemple $\mathbf{x}_i$).
Assumez que

$$\frac{\partial}{\partial a} \max\{0, a\} = \begin{cases} 1 & \text{if } a > 0 \\ 0 & \text{if } a \leq 0 \end{cases}$$

(Cette dernière affirmation n'est pas exactement vraie: à $\mathbf{a} = 0$, la dérivée n'est pas définie. Cependant, pour ce problème, nous allons assumer qu'elle est correcte.)

Solution

Dans le cadre de l'optimisation du SVM un-contre-tous avec pénalité L2 pour la classification multi-classe, il est essentiel de calculer les dérivées de la fonction de perte par rapport aux poids $\mathbf{w}$ afin de mettre à jour ces paramètres lors de l'entraînement par descente de gradient. Après avoir déterminé la dérivée du terme de régularisation dans la question précédente, nous nous concentrons ici sur la dérivée du *hinge loss* par rapport à un poids spécifique w_k^j.

Notre objectif est de calculer :

$$\frac{\partial L_{\text{hinge}}}{\partial w_k^j}$$

et d'exprimer cette dérivée en termes de $x_{i,k}$, tout en fournissant des explications détaillées à chaque étape pour assurer une compréhension approfondie du processus.

Démarche

Étape 1 : Écriture explicite du *hinge loss* individuel

Le *hinge loss* individuel pour un exemple $(\mathbf{x}_i, y_i)$ et une classe j' est défini par :

$$\mathcal{L}\left(\mathbf{w}^{j'}; (\mathbf{x}_i, y_i)\right) = \left(\max\left\{0,\, 2 - t_i^{j'} s_i^{j'}\right\}\right)^2$$

où :

- $t_i^{j'} = \mathbb{1}\{y_i = j'\}$ est le label codé en $\{1, -1\}$:
$$t_i^{j'} = \begin{cases} 1 & \text{si } y_i = j' \\ -1 & \text{si } y_i \neq j' \end{cases}$$
- $s_i^{j'} = \mathbf{w}^{j'} \cdot \mathbf{x}_i = \sum_{l=1}^{p} w_l^{j'} x_{i,l}$ est le score pour la classe j'.

Étape 2 : Identification des termes dépendant de w_k^j

Le poids w_k^j n'apparaît que dans le vecteur de poids $\mathbf{w}^j$ correspondant à la classe j. Par conséquent, pour $j' \neq j$, la dérivée de $\mathcal{L}\left(\mathbf{w}^{j'}; (\mathbf{x}_i, y_i)\right)$ par rapport à w_k^j est nulle :

$$\frac{\partial}{\partial w_k^j} \mathcal{L}\left(\mathbf{w}^{j'}; (\mathbf{x}_i, y_i)\right) = 0 \quad \text{pour} \quad j' \neq j$$

Ainsi, la dérivée totale du *hinge loss* par rapport à w_k^j se simplifie en :

$$\frac{\partial L_{\text{hinge}}}{\partial w_k^j} = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial w_k^j} \mathcal{L}\left(\mathbf{w}^j; (\mathbf{x}_i, y_i)\right)$$

Étape 3 : Calcul de la dérivée de la perte individuelle

Pour calculer $\frac{\partial}{\partial w_k^j} \mathcal{L}\left(\mathbf{w}^j; (\mathbf{x}_i, y_i)\right)$, nous appliquons la règle de la chaîne.

D'abord, exprimons la perte individuelle :

$$\mathcal{L}\left(\mathbf{w}^j; (\mathbf{x}_i, y_i)\right) = \left(\max\{0, z_i\}\right)^2$$

où nous avons défini :

$$z_i = 2 - t_i^j s_i^j$$

avec :

- t_i^j est défini comme précédemment.
- $s_i^j = \mathbf{w}^j \cdot \mathbf{x}_i = \sum_{l=1}^{p} w_l^j x_{i,l}$.

Ensuite, calculons la dérivée en utilisant la règle de la chaîne :

$$\frac{\partial}{\partial w_k^j} \mathcal{L} = 2 \max\{0, z_i\} \cdot \frac{\partial}{\partial w_k^j} (\max\{0, z_i\})$$

Étape 4 : Calcul de la dérivée de $\max\{0, z_i\}$

En utilisant l'assomption donnée :

$$\frac{\partial}{\partial z_i} \max\{0, z_i\} = \begin{cases} 1 & \text{si } z_i > 0 \\ 0 & \text{si } z_i \leq 0 \end{cases}$$

Donc :

$$\frac{\partial}{\partial w_k^j} (\max\{0, z_i\}) = \frac{\partial \max\{0, z_i\}}{\partial z_i} \cdot \frac{\partial z_i}{\partial w_k^j} = \mathbb{I}\{z_i > 0\} \cdot \left(\frac{\partial z_i}{\partial w_k^j}\right)$$

où $\mathbb{I}\{z_i > 0\}$ est la fonction indicatrice valant 1 si $z_i > 0$ et 0 sinon.

Étape 5 : Calcul de $\frac{\partial z_i}{\partial w_k^j}$

Rappelons que :

$$z_i = 2 - t_i^j s_i^j$$

Donc :

$$\frac{\partial z_i}{\partial w_k^j} = -t_i^j \frac{\partial s_i^j}{\partial w_k^j}$$

Or,

$$s_i^j = \sum_{l=1}^{p} w_l^j x_{i,l}$$

Ainsi :

$$\frac{\partial s_i^j}{\partial w_k^j} = x_{i,k}$$

Donc :

$$\frac{\partial z_i}{\partial w_k^j} = -t_i^j x_{i,k}$$

Étape 6 : Combinaison des résultats

En remplaçant dans l'expression de la dérivée :

$$\frac{\partial}{\partial w_k^j} \mathcal{L} = 2 \max\{0, z_i\} \cdot \mathbb{I}\{z_i > 0\} \cdot \left(-t_i^j x_{i,k}\right)$$

Simplifions :

$$\frac{\partial}{\partial w_k^j} \mathcal{L} = -2 \max\{0, z_i\} \cdot t_i^j x_{i,k} \cdot \mathbb{I}\{z_i > 0\}$$

Observant que $\max\{0, z_i\} \cdot \mathbb{I}\{z_i > 0\} = z_i \cdot \mathbb{I}\{z_i > 0\}$, l'expression devient :

$$\frac{\partial}{\partial w_k^j} \mathcal{L} = -2 z_i \cdot t_i^j x_{i,k} \cdot \mathbb{I}\{z_i > 0\}$$

Étape 7 : Expression de la dérivée totale

En revenant à la dérivée totale :

$$\frac{\partial L_{\text{hinge}}}{\partial w_k^j} = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial w_k^j} \mathcal{L}\left(\mathbf{w}^j; (\mathbf{x}_i, y_i)\right)$$

Cela devient :

$$= -\frac{2}{n} \sum_{i=1}^{n} z_i \cdot t_i^j \cdot x_{i,k} \cdot \mathbb{I}\{z_i > 0\}$$

En remplaçant z_i par son expression :

$$\frac{\partial L_{\text{hinge}}}{\partial w_k^j} = -\frac{2}{n} \sum_{i=1}^{n} \left(2 - t_i^j \cdot s_i^j\right) t_i^j \cdot x_{i,k} \cdot \mathbb{I}\left\{2 - t_i^j \cdot s_i^j > 0\right\}$$

Conclusion

La dérivée du *hinge loss* par rapport au poids w_k^j est :

$$\frac{\partial L_{\text{hinge}}}{\partial w_k^j} = -\frac{2}{n} \sum_{i=1}^{n} \left(2 - t_i^j s_i^j\right) t_i^j x_{i,k} \cdot \mathbb{I}\left\{2 - t_i^j s_i^j > 0\right\}$$

Cette expression permet de calculer le gradient du *hinge loss* par rapport à chaque poids individuel, ce qui est essentiel pour la mise à jour des poids lors de l'optimisation par descente de gradient. Elle tient compte des

exemples pour lesquels la marge est violée ($z_i > 0$) et ajuste les poids en conséquence pour améliorer la classification.

En combinant cette dérivée avec celle du terme de régularisation obtenue précédemment :

$$\frac{\partial R}{\partial w_k^j} = C w_k^j$$

le gradient total pour la mise à jour du poids w_k^j est :

$$\frac{\partial L_{\text{total}}}{\partial w_k^j} = \frac{\partial L_{\text{hinge}}}{\partial w_k^j} + \frac{\partial R}{\partial w_k^j} = -\frac{2}{n} \sum_{i=1}^{n} \left(2 - t_i^j s_i^j\right) t_i^j x_{i,k} \cdot \mathbb{I}\left\{2 - t_i^j s_i^j > 0\right\} + C w_k^j$$

Cette formule est fondamentale pour implémenter efficacement l'algorithme de descente de gradient dans le cadre du SVM un-contre-tous avec pénalité L2, assurant une optimisation équilibrée entre la minimisation de la perte sur les données d'entraînement et la régularisation pour éviter le surapprentissage.

Question 4

[5 pts] La méthode Practical.fit utilise le code que vous avez écris cidessus pour entraîner le SVM. Après chaque époque (après avoir passer à travers tous les exemples du jeu de données), SVM.fit calcule la perte et l'exactitude des points d'entraînement et la perte et l'exactitude des points tests.
Faites le graphique de ces quatre quantités en fonction du nombre d'époques, pour $C = 1, 5, 10$. Utilisez comme hyperparamètres 200 époques, un taux d'apprentissage de 0.0001 et une longueur de minibatch de 100.

Vous devriez avoir 4 graphiques, soit un graphique pour chaque quantité, incluant les courbes pour les 3 valeurs de C. Ajoutez ces 4 graphiques dans votre rapport.
Sur la base de ces graphiques, le surapprentissage semble-t-il être un problème pour ce jeu de données et cet algorithme? Expliquez brièvement.

Solution

Dans cette partie, nous analysons l'entraînement d'un SVM un-contre-tous (One-vs-All) pour la classification multi-classe. Nous avons entraîné le modèle en utilisant la méthode Practical.fit pour trois valeurs différentes du paramètre de régularisation $C = 1$, $C = 5$, et $C = 10$. Nous avons utilisé les hyperparamètres suivants :

- Nombre d'époques : 200
- Taux d'apprentissage : 0.0001
- Taille du mini-batch : 100

L'objectif est d'évaluer la performance du modèle à travers quatre mesures :

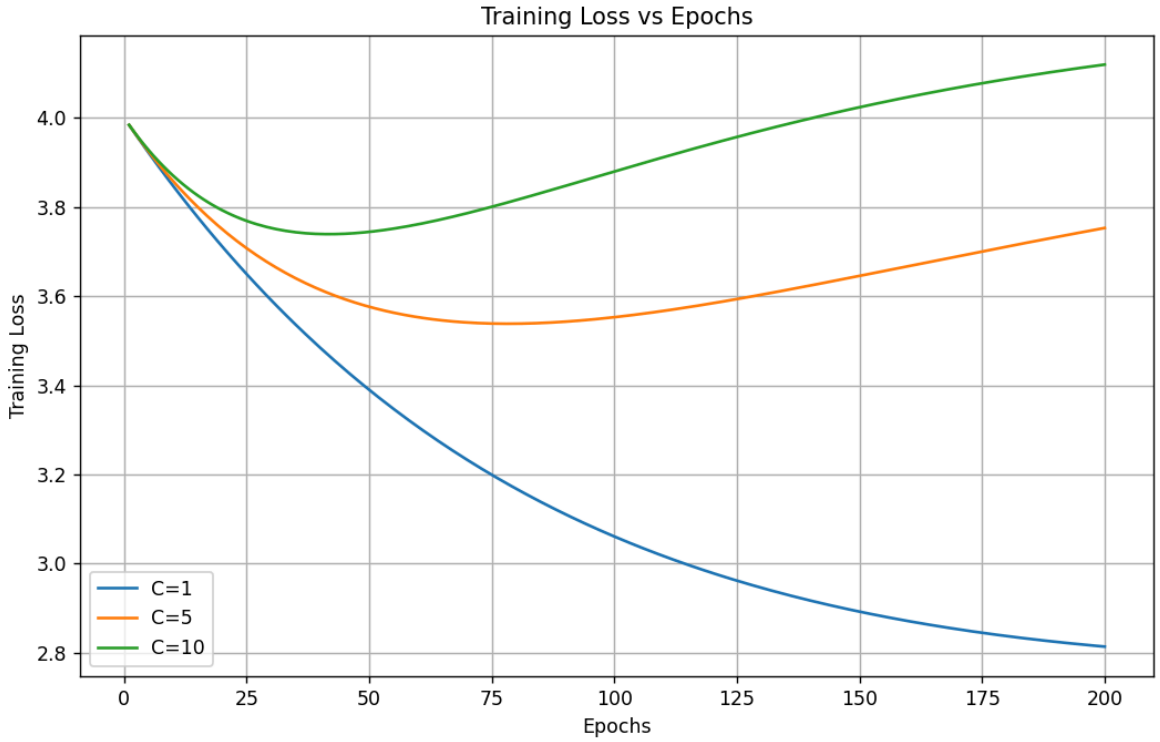
- La perte d'entraînement
- La perte de validation
- L'exactitude sur les données d'entraînement
- L'exactitude sur les données de validation

Nous allons examiner les résultats pour ces quatre quantités en fonction du nombre d'époques et des valeurs de C. Les graphiques obtenus nous permettront de déterminer si le surapprentissage (overfitting) est un problème dans ce contexte.

Résultats

1. Perte d'entraînement

Le premier graphique montre l'évolution de la **perte d'entraînement** en fonction du nombre d'époques pour les trois valeurs de C.

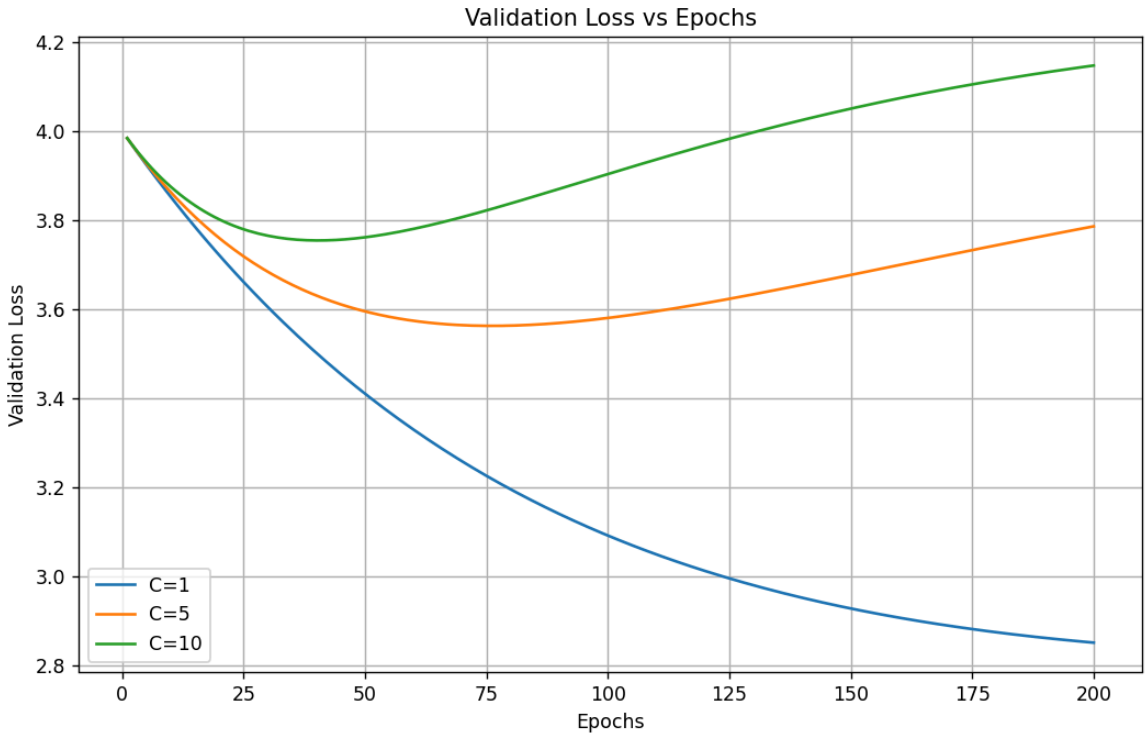


Observations :

- **Pour $C = 1$** : La perte diminue rapidement et continuellement, montrant une convergence progressive vers une solution optimale. Le modèle minimise efficacement l'erreur sur l'ensemble d'entraînement sans signe apparent de surajustement.
- **Pour $C = 5$** : La perte suit une trajectoire descendante au début, mais après environ 75 époques, elle commence à légèrement augmenter. Cela indique que le modèle commence à surajuster les données d'entraînement à partir de ce point.
- **Pour $C = 10$** : La perte initiale diminue, atteignant un minimum rapide après environ 50 époques, puis elle commence à augmenter de manière significative. Ce comportement suggère un surapprentissage important du modèle après une phase initiale de convergence.

2. Perte de validation

Le graphique suivant montre la **perte de validation** en fonction du nombre d'époques.

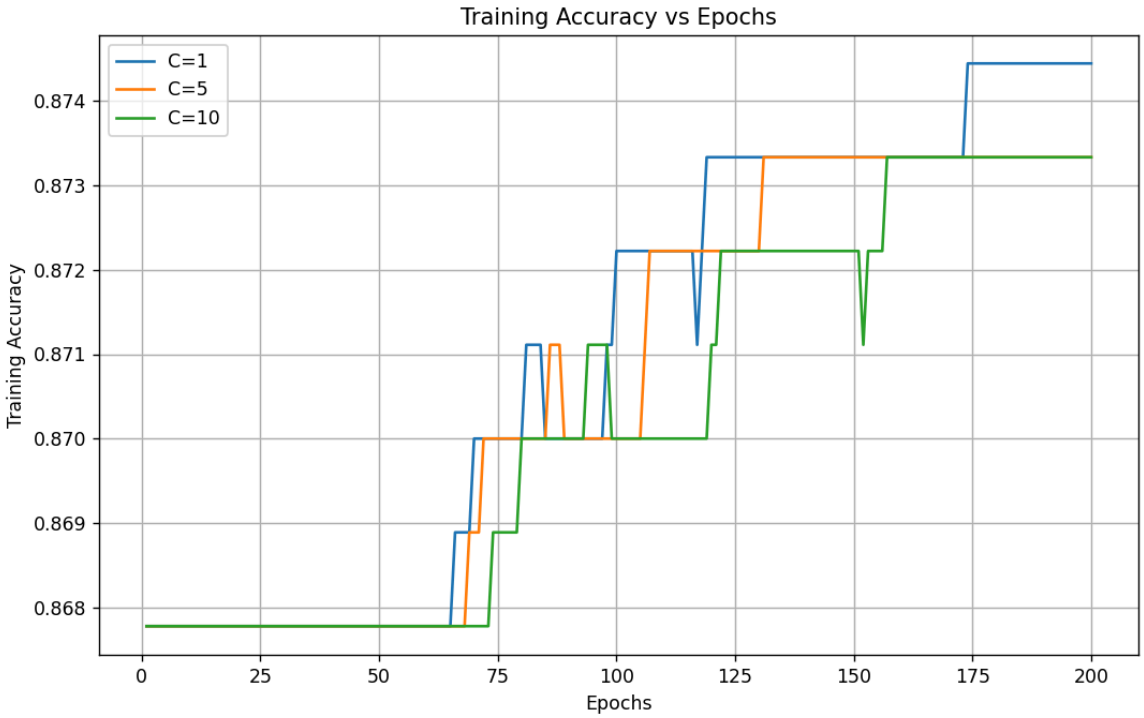


Observations :

- **Pour** $C = 1$: La perte de validation diminue constamment tout au long des époques, ce qui indique une bonne généralisation du modèle sur les données de validation. Ce comportement suggère que le modèle n'est pas en surapprentissage avec cette valeur de C.
- **Pour** $C = 5$: La perte diminue dans les premières époques, atteignant un minimum autour de 75 époques, puis commence à augmenter lentement. Cela peut indiquer un début de surapprentissage à partir de ce point.
- **Pour** $C = 10$: La perte suit une tendance similaire à celle de $C = 5$ mais de façon plus marquée. Elle diminue initialement, atteint un minimum vers les 50 premières époques, puis augmente de manière plus significative, montrant un surapprentissage plus prononcé pour cette valeur de C.

3. Exactitude d'entraînement

Le graphique ci-dessous montre l'évolution de l'**exactitude sur les données d'entraînement**.

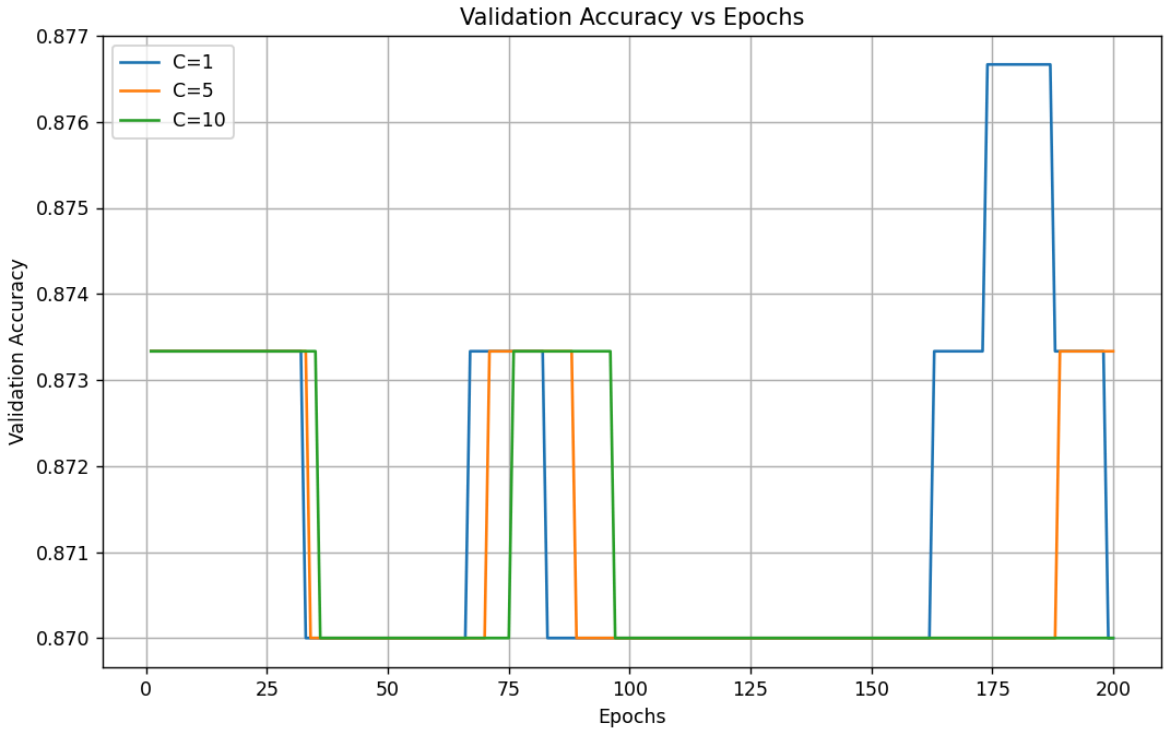


Observations :

- **Pour $C = 1$** : L'exactitude d'entraînement augmente rapidement à partir de la 60e époque et continue de s'améliorer de manière stable, atteignant une valeur proche de 0.874 après 150 époques. Il y a moins de fluctuation dans l'exactitude par rapport aux valeurs plus élevées de C.
- **Pour $C = 5$** : L'exactitude commence également à augmenter vers la 65e époque, mais il y a des fluctuations visibles dans la progression, avec des moments de stagnation et d'amélioration. L'exactitude atteint une valeur similaire à celle de $C = 1$ à la fin des 200 époques.
- **Pour $C = 10$** : Comme pour $C = 5$, l'exactitude augmente à partir de la 70e époque avec des fluctuations notables. Cependant, les progrès sont plus instables, et l'exactitude semble légèrement inférieure ou similaire à celle obtenue pour $C = 5$ et $C = 1$.
- **Fluctuations visibles** : Contrairement aux valeurs de C plus faibles, les valeurs plus élevées de $C = 5$ et $C = 10$ présentent davantage de variations au cours de l'entraînement, ce qui pourrait être un signe d'instabilité dans l'apprentissage du modèle.

4. Exactitude de validation

Le dernier graphique présente l'**exactitude sur les données de validation**.



Observations :

- **Pour $C = 1$** : L'exactitude de validation est relativement stable sur la majorité des époques. Elle reste autour de 0.873, avec une légère augmentation vers la fin (aux alentours de la 175e époque) où elle atteint environ 0.876 avant de redescendre.
- **Pour $C = 5$** : Le comportement de l'exactitude est similaire à celui de $C = 1$, bien qu'elle soit plus fluctuante. Elle augmente légèrement après la 150e époque, mais finit par revenir au même niveau qu'au début, autour de 0.874.
- **Pour $C = 10$** : L'exactitude de validation est beaucoup plus stable et ne montre pratiquement aucun changement au fil des époques. Elle reste très proche de 0.870 tout au long de l'entraînement, sans signe de progression.
- **Fluctuations visibles** : Pour $C = 1$ et $C = 5$, on observe des augmentations notables de l'exactitude vers la fin de l'entraînement, mais ces améliorations ne sont pas durables et redescendent rapidement.

Conclusion sur le surapprentissage

Sur la base des graphiques, il est possible de conclure que le surapprentissage est un problème pour les plus grandes valeurs du paramètre de régularisation C. En particulier :

- **Pour $C = 1$**, le modèle semble bien généraliser aux données de validation, avec une perte de validation en baisse constante et une exactitude qui reste relativement stable. Cela suggère que le modèle n'est pas en surapprentissage avec cette valeur de C.
- **Pour $C = 5$ et $C = 10$**, on observe un comportement caractéristique du surapprentissage. La perte d'entraînement continue de diminuer, indiquant que le modèle s'adapte de plus en plus aux données d'entraînement. Cependant, la perte de validation commence à augmenter après un certain point, et l'exactitude de validation stagne ou diminue, ce qui montre que le modèle ne parvient pas à bien généraliser aux nouvelles données.

Cela indique que des valeurs plus élevées de C, qui réduisent la régularisation, entraînent un ajustement trop spécifique aux données d'entraînement, au détriment de la performance sur les données de validation. Par conséquent, une régularisation plus forte (comme avec $C = 1$) est préférable pour maintenir un bon équilibre entre performance et généralisation, et éviter le surapprentissage.