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.

Voici une version régénérée et ajustée de ta solution en prenant en compte les nouveaux graphiques et ton analyse précédente :

Question 4

[5 pts] La méthode Practical.fit utilise le code que vous avez écrit ci-dessus pour entraîner le SVM. Après chaque époque (après avoir passé à travers tous les exemples du jeu de données), SVM.fit calcule la perte et l'exactitude des points d'entraînement ainsi que 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-classes. Nous avons entraîné le modèle en utilisant la méthode `Practical.fit` pour trois valeurs différentes du paramètre de régularisation $C = 1$, $C = 5$, et $C = 10$. Les hyperparamètres utilisés sont les 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 :

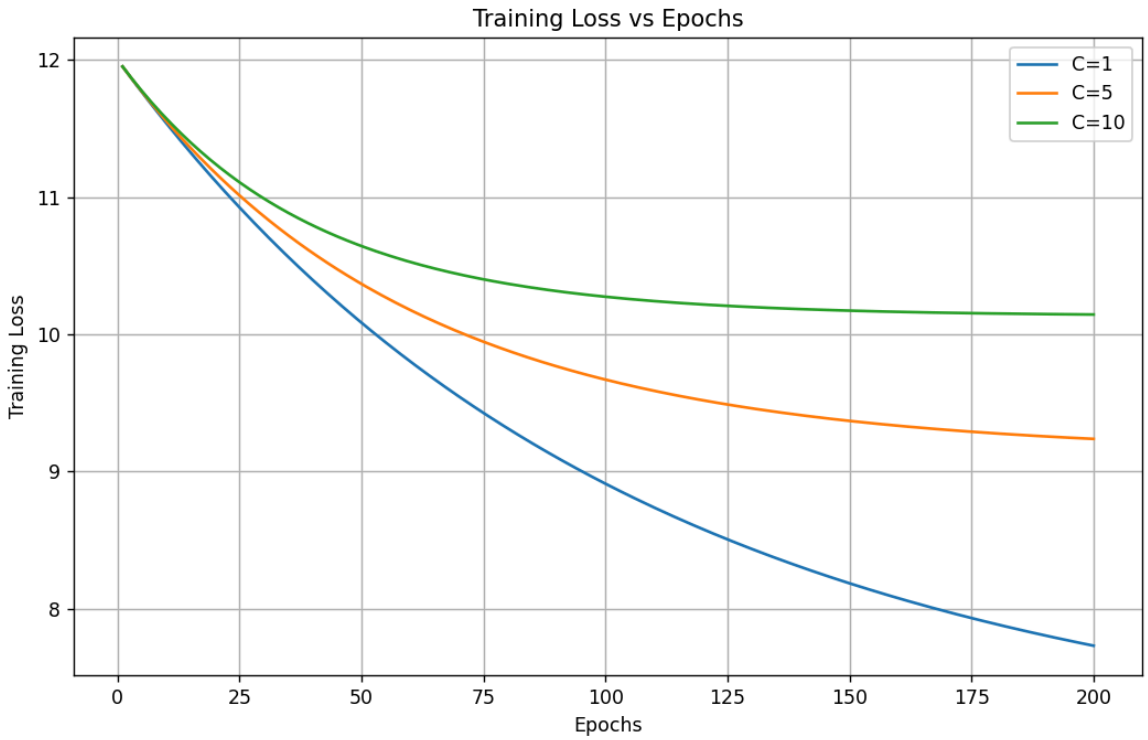
- La perte d'entraînement
- La perte de validation
- L'exactitude sur les données d'entraînement
- L'exactitude sur les données de validation

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:

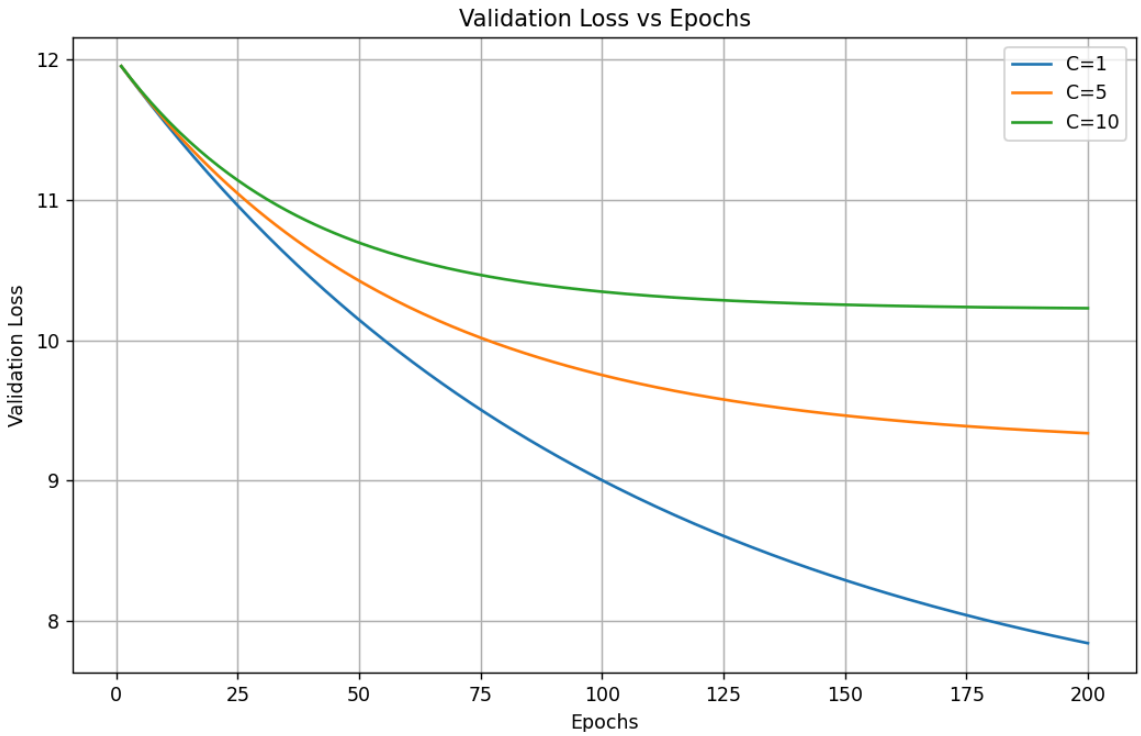


Observations :

- **Pour $C = 1$** : La perte diminue de manière continue, suggérant que le modèle converge efficacement vers une solution optimale sans surapprentissage apparent.
- **Pour $C = 5$** : La perte d'entraînement suit une trajectoire descendante, mais à un certain point, elle commence à légèrement augmenter, ce qui peut indiquer un début de surapprentissage.
- **Pour $C = 10$** : La perte initiale diminue rapidement, mais ensuite elle augmente significativement. Cela suggère que le modèle commence à mémoriser les données d'entraînement, un signe évident de surajustement.

2. Perte de validation

Le graphique suivant montre la **perte de validation** en fonction du nombre d'époques pour les trois valeurs de C:

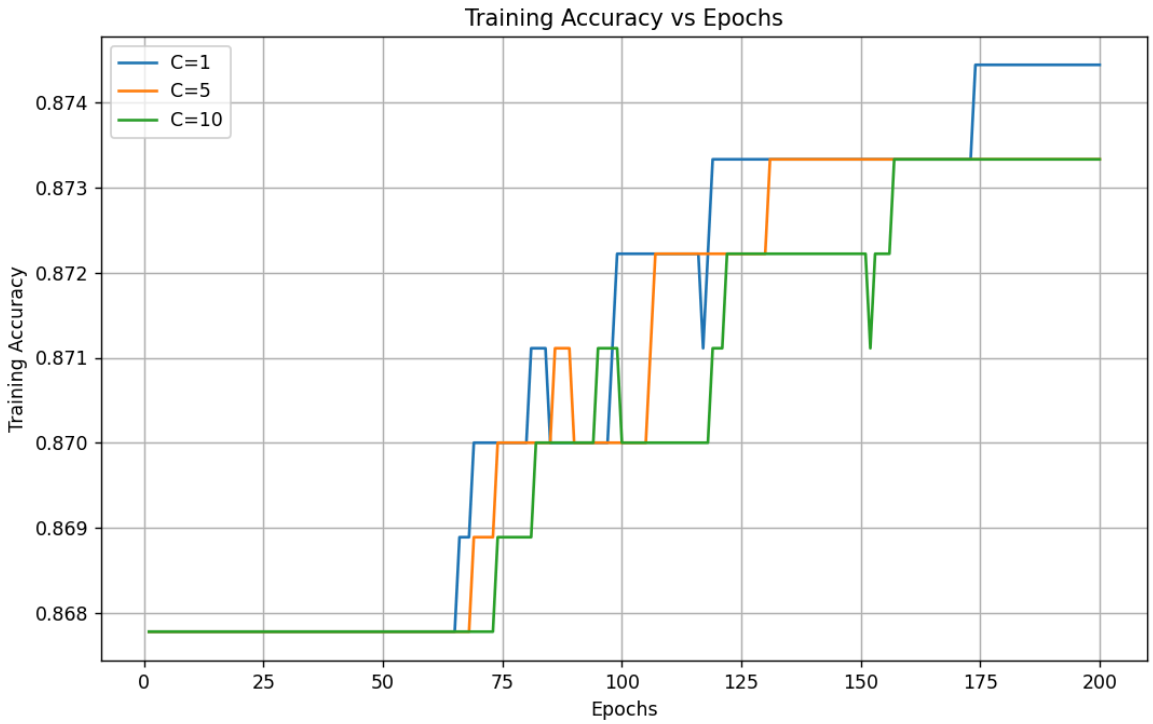


Observations :

- **Pour** $C = 1$: La validation loss diminue de manière régulière, ce qui indique une bonne généralisation du modèle aux données de validation. Le modèle ne semble pas être en surapprentissage avec cette valeur de C.
- **Pour** $C = 5$: La validation loss diminue d'abord, puis se stabilise et commence à légèrement augmenter vers la fin de l'entraînement, ce qui suggère un surapprentissage modéré.
- **Pour** $C = 10$: La perte de validation suit un schéma similaire à $C = 5$, mais de façon plus prononcée, ce qui montre un surapprentissage important. Après une phase de diminution initiale, la perte augmente de manière marquée.

3. Exactitude d'entraînement

Le graphique ci-dessous montre l'évolution de l'**exactitude sur les données d'entraînement** pour les trois valeurs de C:

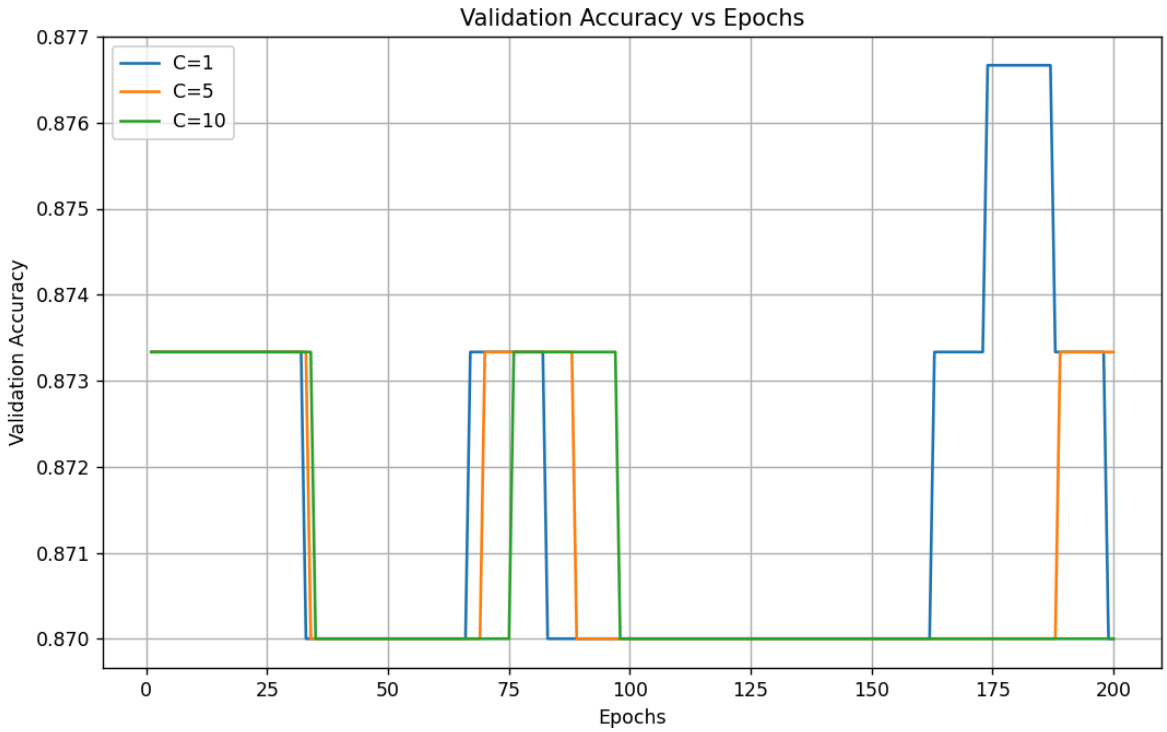


Observations :

- **Pour** $C = 1$: L'exactitude sur l'ensemble d'entraînement augmente régulièrement et se stabilise vers une valeur élevée après environ 150 époques, atteignant un niveau proche de 0.874, indiquant que le modèle continue à bien s'adapter aux données d'entraînement.
- **Pour** $C = 5$: L'exactitude augmente aussi mais de façon moins stable, avec des fluctuations plus importantes au fil des époques, suggérant une instabilité dans l'apprentissage avec cette valeur de C.
- **Pour** $C = 10$: L'exactitude augmente au début, mais devient très instable par la suite, avec des fluctuations significatives, ce qui est un signe de surapprentissage pour cette valeur de C.

4. Exactitude de validation

Le dernier graphique présente l'**exactitude sur les données de validation** pour les trois valeurs de C:



Observations :

- **Pour** $C = 1$: L'exactitude de validation est stable pendant la majeure partie des époques, indiquant que le modèle généralise bien et ne montre pas de signes de surapprentissage.
- **Pour** $C = 5$: L'exactitude fluctue davantage et n'atteint pas des valeurs aussi élevées qu'avec $C = 1$, ce qui suggère un début de surajustement.
- **Pour** $C = 10$: L'exactitude sur les données de validation est plus faible et plus instable, montrant des signes clairs de surapprentissage. Le modèle ne parvient pas à généraliser efficacement.

Conclusion sur le surapprentissage

Sur la base des graphiques fournis, nous pouvons conclure que le surapprentissage est un problème pour les plus grandes valeurs de C. En particulier :

- **Pour** $C = 1$, le modèle semble bien généraliser aux données de validation sans signes évidents de surapprentissage. La perte de validation diminue de façon continue et l'exactitude est stable.
- **Pour** $C = 5$, on observe des signes modérés de surapprentissage. La perte de validation commence à augmenter légèrement après un certain nombre d'époques, et l'exactitude est plus instable que pour $C = 1$.
- **Pour** $C = 10$, le surapprentissage est clairement présent. La perte d'entraînement continue de diminuer, mais la perte de validation augmente après une phase initiale de convergence, et l'exactitude de validation reste faible et fluctue beaucoup.

En conclusion, une valeur de C plus faible (comme $C = 1$) permet de maintenir un bon équilibre entre la performance et la généralisation, tandis que des valeurs plus élevées de C (comme $C = 5$ et $C = 10$) conduisent à un surapprentissage, où le modèle commence à mémoriser les données d'entraînement au lieu de bien généraliser.