Introduction aux réseaux de neurones – exercices

Rappel : Neurones de sorties pour un problème de classification multi-classes
Soit un problème à C classes.

1. Pour un exemple d'apprentissage (x, y), on représente chaque classe par un entier
$$y \in \{1, \dots, C\}$$
2. On converti y sous la forme d'un vecteur «one-hot» $\mathbf{y} \in \mathbb{R}^C$, possédant la valeur 1 à l'index correspondant à y, et les valeurs 0 autrement :
$$y = 0 \mapsto \mathbf{y} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad y = 1 \mapsto \mathbf{y} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$
3. La couche de sortie sera donnée un vecteur $\hat{\mathbf{y}}$ obtenu en appliquant la fonction d'activation «softmax» aux valeurs $\mathbf{a} \in \mathbb{R}^C$ propagées par les couches précédentes :
$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_C \end{bmatrix} \quad \text{avec } \hat{y}_i = \text{softmax}(a_i) = \frac{e^{a_i}}{\sum_{j=1}^{C} e^{a_j}}$$
On interprète le vecteur $\mathbf{y}$ comme une distribution de probabilité sur les classes.
4. Lors de l'optimisation du réseau, on minimise la perte du négatif log-vraisemblance :
$$L_{\text{nlv}}(\hat{\mathbf{y}}, y) = -\ln(\hat{y}_y),$$
où $\hat{y}_y$ corresponds à la sortie d'index y du réseau (la probabilité associée à la classe de l'exemple).

Question 1. Calculer $\nabla_{\mathbf{a}} L_{\text{nlv}}(\hat{y}_y, y)$, c'est-à-dire le gradient qui modifiera les poids associés à la couche de sortie. Pour ce faire, procédons en trois étapes

(a) Calculer la dérivée partielle associée au neurone de sortie correspondant à la bonne classe $(y = i)$:
$$\frac{\partial}{\partial a_y} L_{\text{nlv}}(\hat{y}_y, y) = \frac{\partial}{\partial a_y} \ln \left[\frac{1}{\text{softmax}(a_y)} \right].$$

(b) Calculer la dérivée partielle associée au neurone de sortie correspondant à la bonne classe $(y \neq i)$:
$$\frac{\partial}{\partial a_i} L_{\text{nlv}}(\hat{y}_y, y) = \frac{\partial}{\partial a_i} \ln \left[\frac{1}{\text{softmax}(a_y)} \right], \text{ avec } a_i \neq a_y \,.$$

(c) À partir des réponses (a) et (b), déduire le gradient $\nabla_{\mathbf{a}} L_{\text{nlv}}(\hat{y}_y, y)$.

Question 2. Considérons un réseau de neurones qui reçoit en entrée des images en teintes de gris représentées par des matrices de 20×20 pixels. Chacune de ces images illustre un des fruits suivant : banane, pomme, poire ou ananas. Il s'agit donc d'un problème de classification multi-classes ; le réseau possède quatre neurones de sorties.

(a) Si le réseau possède une seule couche cachée pleinement connectée de n neurones, combien de paramètres devrons-nous optimiser par descente de gradient ?
   — Si on n'utilise aucun paramètre de biais, ni pour la couche cachée et ni pour la couche de sortie ?
   — Si on inclut les paramètres de biais pour la couche cachée et pour la couche de sortie ?

(b) Si le réseau possède une première couche de m filtres de convolutions de taille 5×5, suivie d'une couche cachée pleinement connectée de n neurones, et d'aucun paramètre de biais, combien de paramètres devrons-nous optimiser par descente de gradient ?

(c) Si le réseau possède une première couche de m filtres de convolutions de taille 5×5, suivit d'une couche de *MaxPooling* avec une taille de pas de 4, suivit d'une couche cachée pleinement connectée de n neurones, et d'aucun paramètre de biais, combien de paramètres devrons-nous optimiser par descente de gradient ?

(d) Enfin, considérons l'architecture suivante :
   — Entrée du réseau : images de tailles 20×20 ;
   — 1ère couche : m_1 filtres de convolutions de taille 5×5, sans biais.
   — *MaxPooling* avec taille de pas 2
   — 2e couche : m_2 filtres de convolutions de taille $3 \times 3 \times m_1$, sans biais.
   — *MaxPooling* avec taille de pas 2
   — 3e couche : n_1 neurones pleinement connectés, sans biais
   — 4e couche : n_2 neurones pleinement connectés, sans biais
   — 5e couche : 4 neurones de sortie, avec biais.

Question 3. Reprendre la question 2(a-d) en considérant cette fois-ci que les images sont en couleurs. Le réseau reçoit donc en entrée des matrices de taille $20 \times 20 \times 3$.