



Examen formatif

GRO721  
Réseaux de neurones convolutifs en traitement d'images

Génie robotique  
Faculté de génie  
Université de Sherbrooke

Hiver 2022

Copyright ©2022, Faculté de génie  
Université de Sherbrooke

*(Cette page est laissée vide intentionnellement)*

|            |            |
|------------|------------|
| GRO-721 #1 | GRO-721 #2 |
| /40        |            |

*S.V.P. Ne rien inscrire dans les grilles réservées à la compilation de l'évaluation*

## Question 1

Répondre par vrai ou faux aux affirmations suivantes.

1. Un réseau effectuant une tâche de segmentation sémantique est normalement entraîné avec une fonction de coût d'erreur quadratique moyenne.
2. Un réseau avec des connexions résiduelles nécessite généralement moins d'itérations durant son entraînement par rapport à un réseau équivalent sans connexions résiduelles.
3. Il est possible de remplacer une couche linéaire par une couche convective.
4. Un réseau U-Net doit absolument utiliser une couche de softmax en sortie.

- 1) Faux. Il s'agit d'une tâche de classification pour chaque pixel, donc on utilise une fonction de coût d'entropie croisée.
- 2) Vrai. Le résidu permet au gradient de se propager plus efficacement, ce qui réduit le phénomène de disparition du gradient, et donc accélère la convergence du réseau.
- 3) Vrai. Voir procédural 1, question 4.
- 4) Faux. Il est possible également d'utiliser une fonction sigmoïde pour une classification binaire.

*(Espace supplémentaire si nécessaire pour la question 1)*

|            |            |
|------------|------------|
| GRO-721 #1 | GRO-721 #2 |
| /40        |            |

S.V.P. Ne rien inscrire dans les grilles réservées à la compilation de l'évaluation

## Question 2

### Section A [20 pts]

Soit la couche convulsive suivante avec foulées horizontale et verticale de  $S_W = 3$  et  $S_H = 3$ , remplissages horizontal et vertical de  $P_W = 2$  et  $P_H = 2$  et le noyau de  $\mathbf{W} \in \mathbb{R}^{1 \times 3 \times 3}$  suivant:

$$\mathbf{W} = \begin{bmatrix} +1 & +2 & -1 \\ +0 & -1 & +1 \\ -2 & +0 & -3 \end{bmatrix}$$

Il n'y a pas de biais. Le tenseur d'entrée ( $\mathbf{X} \in \mathbb{R}^{1 \times 5 \times 5}$ ) est le suivant:

$$\mathbf{X} = \begin{bmatrix} +4 & +1 & -2 & -3 & +4 \\ +0 & -3 & +3 & +1 & -2 \\ -2 & +2 & +3 & +0 & +2 \\ +2 & +0 & +1 & +1 & +1 \\ -3 & +3 & -1 & -1 & +2 \end{bmatrix}$$

Calculer le tenseur de sortie  $\mathbf{Y}$  en appliquant cette couche convulsive au vecteur d'entrée.

### Section B [20 pts]

Soit la couche convulsive transposée avec foulées horizontale et vertical de  $S_W = 2$  et  $S_H = 2$ , aucun remplissage, et le noyau de  $\mathbf{W} \in \mathbb{R}^{1 \times 2 \times 2}$  suivant:

$$\mathbf{W} = \begin{bmatrix} +2 & +1 \\ +3 & +1 \end{bmatrix}$$

Il n'y a pas de biais. Le tenseur d'entrée ( $\mathbf{X} \in \mathbb{R}^{1 \times 3 \times 3}$ ) est le suivant:

$$\mathbf{X} = \begin{bmatrix} +2 & -2 & +0 \\ +1 & -1 & +0 \\ +3 & +1 & +1 \end{bmatrix}$$

Calculer le tenseur de sortie  $\mathbf{Y}$  en appliquant cette couche convulsive au vecteur d'entrée.

### Section A

$$\mathbf{X} = \begin{bmatrix} +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 \\ +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 \\ +0 & +0 & +4 & +1 & -2 & -3 & +4 & +0 & +0 \\ +0 & +0 & +0 & -3 & +3 & +1 & -2 & +0 & +0 \\ +0 & +0 & -2 & +2 & +3 & +0 & +2 & +0 & +0 \\ +0 & +0 & +2 & +0 & +1 & +1 & +1 & +0 & +0 \\ +0 & +0 & -3 & +3 & -1 & -1 & +2 & +0 & +0 \\ +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 \\ +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 \end{bmatrix}$$

(Espace supplémentaire si nécessaire pour la question 2)

$$Y_{11} = (-3)(+4) = -12$$

$$Y_{12} = (-2)(+1) + (+0)(-2) + (-3)(-3) = +7$$

$$Y_{13} = (-2)(+4) = -8$$

$$Y_{21} = (-1)(+0) + (+1)(-2) + (-3)(+2) = -8$$

$$Y_{22} = (+1)(-3) + (+2)(+3) + (-1)(+1) + (+0)(+2) + (-1)(+3) + (+1)(+0) + (-2)(+0) + (+0)(+1) + (-3)(+1) = -4$$

$$Y_{23} = (-2)(+1) + (+2)(+0) + (+1)(-2) = -4$$

$$Y_{31} = (-1)(-3) = +3$$

$$Y_{32} = (+1)(+3) + (+2)(-1) + (-1)(-1) = +2$$

$$Y_{33} = (+1)(+2) = +2$$

$$\mathbf{Y} = \begin{bmatrix} -12 & +7 & -8 \\ -8 & -4 & -4 \\ +3 & +2 & +2 \end{bmatrix}$$

## Section B

$$+2 \begin{bmatrix} +2 & +1 \\ +3 & +1 \end{bmatrix} = \begin{bmatrix} +4 & +2 \\ +6 & +2 \end{bmatrix}$$

$$-2 \begin{bmatrix} +2 & +1 \\ +3 & +1 \end{bmatrix} = \begin{bmatrix} -4 & -2 \\ -6 & -2 \end{bmatrix}$$

$$+0 \begin{bmatrix} +2 & +1 \\ +3 & +1 \end{bmatrix} = \begin{bmatrix} +0 & +0 \\ +0 & +0 \end{bmatrix}$$

$$+1 \begin{bmatrix} +2 & +1 \\ +3 & +1 \end{bmatrix} = \begin{bmatrix} +2 & +1 \\ +3 & +1 \end{bmatrix}$$

$$-1 \begin{bmatrix} +2 & +1 \\ +3 & +1 \end{bmatrix} = \begin{bmatrix} -2 & -1 \\ -3 & -1 \end{bmatrix}$$

$$+0 \begin{bmatrix} +2 & +1 \\ +3 & +1 \end{bmatrix} = \begin{bmatrix} +0 & +0 \\ +0 & +0 \end{bmatrix}$$

$$+3 \begin{bmatrix} +2 & +1 \\ +3 & +1 \end{bmatrix} = \begin{bmatrix} +6 & +3 \\ +9 & +3 \end{bmatrix}$$

$$+1 \begin{bmatrix} +2 & +1 \\ +3 & +1 \end{bmatrix} = \begin{bmatrix} +2 & +1 \\ +3 & +1 \end{bmatrix}$$

$$+1 \begin{bmatrix} +2 & +1 \\ +3 & +1 \end{bmatrix} = \begin{bmatrix} +2 & +1 \\ +3 & +1 \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} +4 & +2 & -4 & -2 & +0 & +0 \\ +6 & +2 & -6 & -2 & +0 & +0 \\ +2 & +1 & -2 & -1 & +0 & +0 \\ +3 & +1 & -3 & -1 & +0 & +0 \\ +6 & +3 & +2 & +1 & +2 & +1 \\ +9 & +3 & +3 & +1 & +3 & +1 \end{bmatrix}$$

|            |            |
|------------|------------|
| GRO-721 #1 | GRO-721 #2 |
|            | /40        |

*S.V.P. Ne rien inscrire dans les grilles réservées à la compilation de l'évaluation*

## Question 3

Dans une tâche de localisation d'image, la cible est représentée par une boîte dont le coin supérieur gauche se situe au pixel (57, 23) et le coin inférieur droit est au pixel (201, 354). Nous avons trois prédictions de boîtes, dont les coins supérieur gauche (CSG) et inférieur droit (CID) sont:

1. CSG: (24, 10), CID: (220, 400)
2. CSG: (80, 70), CID: (180, 300)
3. CSG: (100, 30), CID: (205, 370)

Quelle prédiction est la plus performante en terme d'intersection sur union? Donnez les détails de vos calculs.

Boîte 1:

$$\text{Intersection: } (201 - 57) \times (354 - 23) = 47664$$

$$\text{Union: } (220 - 24) \times (400 - 10) = 76440$$

$$\text{Intersection sur Union: } IoU = 47664 / 76440 = 0.6235$$

Boîte 2:

$$\text{Intersection: } (180 - 80) \times (300 - 70) = 23000$$

$$\text{Union: } (201 - 57) \times (354 - 23) = 47664$$

$$\text{Intersection sur Union: } IoU = 23000 / 47664 = 0.4825$$

Boîte 3:

$$\text{Intersection: } (201 - 100) \times (354 - 30) = 32724$$

$$\text{Union: } (205 - 57) \times (370 - 23) - (100 - 57) \times (370 - 354) - (205 - 201) \times (30 - 23) = 50640 \text{ Intersection sur Union:}$$

$$IoU = 32724 / 50640 = 0.6462$$

La boîte 3 offre le meilleur IoU, et donc la prédiction la plus proche de la cible.

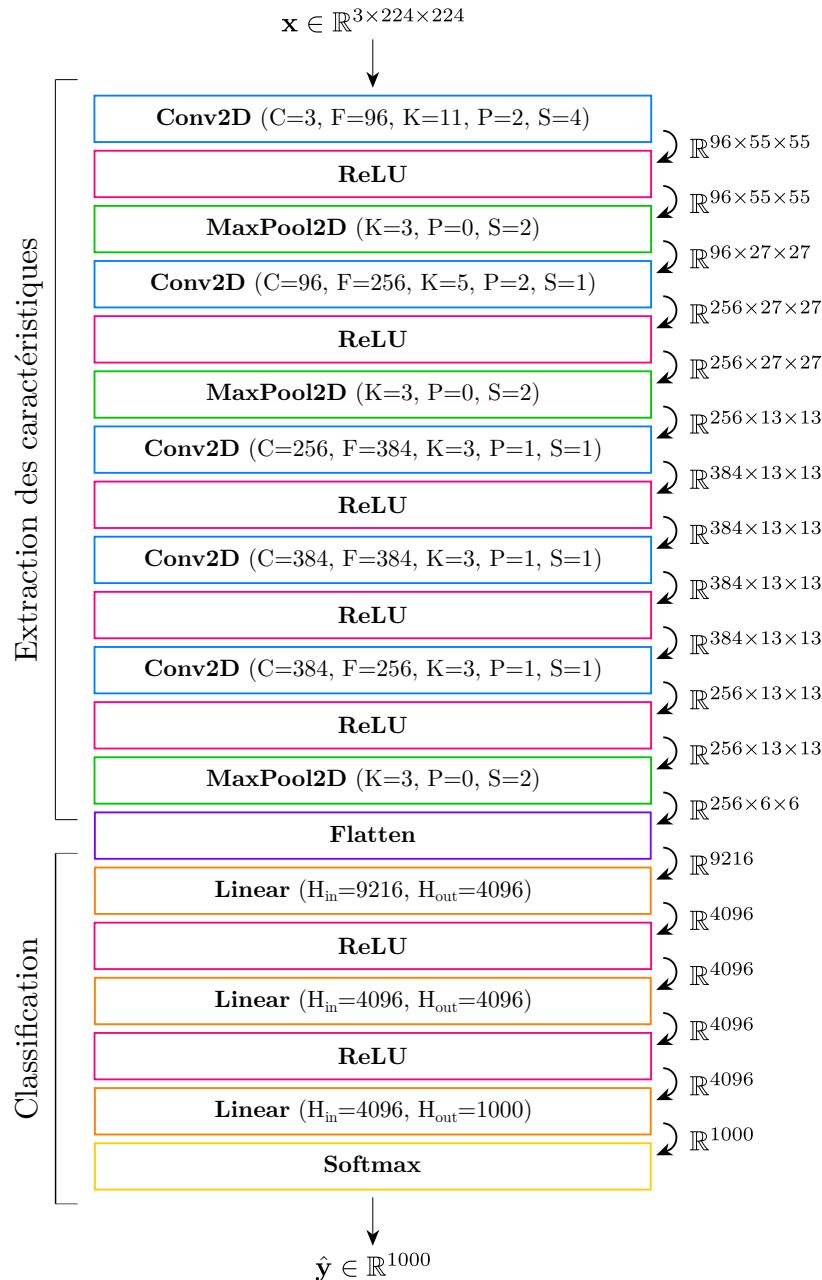
(Espace supplémentaire si nécessaire pour la question 3)

|            |            |
|------------|------------|
| GRO-721 #1 | GRO-721 #2 |
| /40        |            |

S.V.P. Ne rien inscrire dans les grilles réservées à la compilation de l'évaluation

## Question 4

Combien de paramètres doivent être appris lors de l'entraînement d'AlexNet? Nous assumons que les couches convolutives et linéaires possèdent des biais.



(Espace supplémentaire si nécessaire pour la question 4)

Couche Conv 1:  $3 \times 96 \times 11 \times 11 + 96 = 34944$

Couche Conv 2:  $96 \times 256 \times 5 \times 5 + 256 = 614656$

Couche Conv 3:  $256 \times 384 \times 3 \times 3 + 384 = 885120$

Couche Conv 4:  $384 \times 384 \times 3 \times 3 + 384 = 1327488$

Couche Conv 5:  $384 \times 256 \times 3 \times 3 + 256 = 884992$

Couche Lin 1:  $9216 \times 4096 + 4096 = 37752832$

Couche Lin 2:  $4096 \times 4096 + 4096 = 16781312$

Couche Lin 3:  $4096 \times 1000 + 1000 = 4097000$

Total:  $34944 + 614656 + 885120 + 1327488 + 884992 + 37752832 + 16781312 + 4097000 = 62378344$

|            |            |
|------------|------------|
| GRO-721 #1 | GRO-721 #2 |
|            | /40        |

*S.V.P. Ne rien inscrire dans les grilles réservées à la compilation de l'évaluation*

## Question 5

Écrivez le code Python qui permettra de réaliser le réseau suivant avec la classe `torch.nn.Module`. Vous devez écrire la fonction d'initialisation du réseau et la fonction pour la propagation vers l'avant. Le réseau reçoit un tenseur de  $N \times 3 \times 128 \times 128$  en entrée. Il doit effectuer une classification parmi 10 classes. Une seule classe peut être active en sortie.

---

Convolution 2D – 16 Noyaux  $3 \times 3$ , Foulée de 1, Remplissage de 1

Mise en commun maximale –  $2 \times 2$ , Foulée de 2

Rectificateur

---

Convolution 2D – 32 Noyaux  $3 \times 3$ , Foulée de 1, Remplissage de 1

Mise en commun maximale –  $2 \times 2$ , Foulée de 2

Rectificateur

---

Convolution 2D – 32 Noyaux  $3 \times 3$ , Foulée de 1, Remplissage de 1

Mise en commun maximale –  $2 \times 2$ , Foulée de 2

Rectificateur

---

Convolution 2D – 32 Noyaux  $3 \times 3$ , Foulée de 1, Remplissage de 1

Mise en commun maximale –  $2 \times 2$ , Foulée de 2

Rectificateur

---

Convolution 2D – 10 Noyaux  $8 \times 8$ , Foulée de 1, Remplissage de 0

Softmax

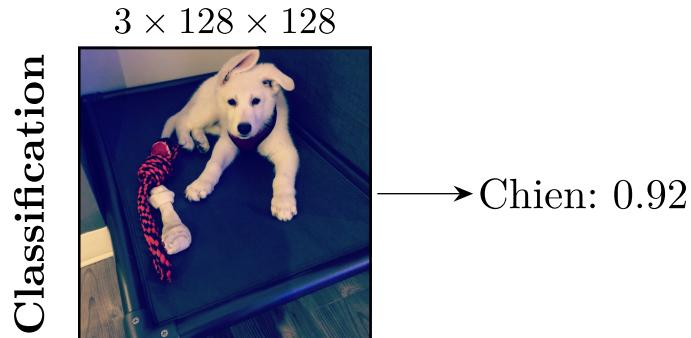
---

|            |            |
|------------|------------|
| GRO-721 #1 | GRO-721 #2 |
|            | /40        |

S.V.P. Ne rien inscrire dans les grilles réservées à la compilation de l'évaluation

## Question 6

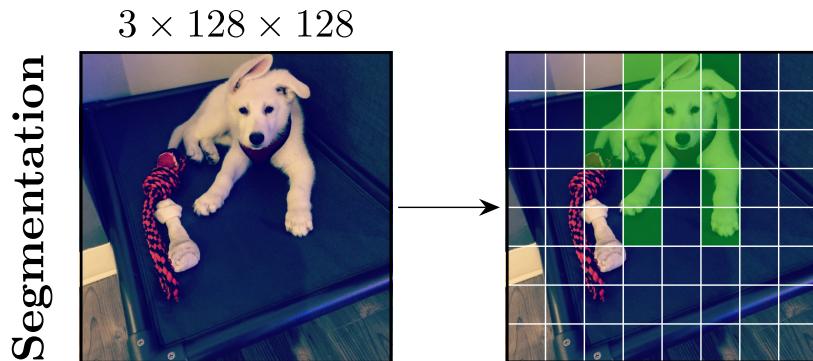
Un réseau de neurones est entraîné pour effectuer une classification à partir d'une image RGB de  $128 \times 128$  pixels. Le réseau sort une sortie de 0 si aucun chien n'est présent dans l'image, et une sortie qui tend vers 1 si un chien est présent:



Un réseau convolutif est utilisé pour la classification. La notation suivante est utilisée: Conv2D ( $F = 16, K = 3, P = 1, S = 1$ ) qui signifie une couche convolutive en 2D avec 16 noyaux de dimensions  $3 \times 3$ , avec un remplissage de  $P = 1$  et une foulée de  $S = 1$ . De manière semblable, l'expression MaxPool2D ( $K = 2, S = 2$ ) signifie une couche de mise en commun maximale avec un noyau de dimensions  $2 \times 2$ , et une foulée de  $S = 2$ .

- 
- Conv2D ( $F = 16, K = 3, P = 1, S = 1$ ) + ReLU + MaxPool2D ( $K = 4, S = 4$ )
  - Conv2D ( $F = 32, K = 3, P = 1, S = 1$ ) + ReLU + MaxPool2D ( $K = 4, S = 4$ )
  - Conv2D ( $F = 64, K = 3, P = 1, S = 1$ ) + ReLU + MaxPool2D ( $K = 4, S = 4$ )
  - Conv2D ( $F = 128, K = 3, P = 1, S = 1$ ) + ReLU + MaxPool2D ( $K = 2, S = 2$ )
- 
- Conv2D ( $F = 128, K = 1, P = 0, S = 1$ ) + Sigmoid
- 

Vous entraînez ce réseau sur un grand lot d'images et vous obtenez d'excellentes performances. Vous désirez ensuite réutiliser une partie de ce réseau pour faire du transfert d'apprentissage et effectuer une segmentation pour définir les zones qui contiennent une image de chien sur une grille de dimensions  $8 \times 8$ , tel qu'illustré ci-dessous.



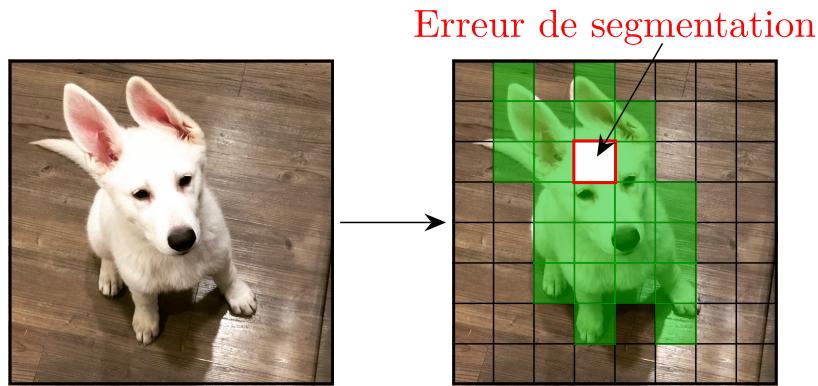
(Espace supplémentaire si nécessaire pour la question 6)

### Section A [30 pts]

Quelle serait l'architecture du nouveau réseau de neurones, et quelles couches préalablement entraînées durant la classification seraient réutilisées?

### Section B [10 pts]

Vous remarquez que pour certaines images la segmentation n'est pas idéale. Par exemple, dans l'image ci-dessous une des cases de la grille n'est pas segmentée correctement. Qu'est-ce qui dans l'architecture du réseau pourrait favoriser ce genre d'erreur?



### Section A

On conserve les deux premières couches suivantes:

---


$$\begin{aligned} \text{Conv2D } (F = 16, K = 3, P = 1, S = 1) + \text{ReLU} + \text{MaxPool2D } (K = 4, S = 4) \\ \text{Conv2D } (F = 32, K = 3, P = 1, S = 1) + \text{ReLU} + \text{MaxPool2D } (K = 4, S = 4) \end{aligned}$$


---

Après ces couches on obtient un tenseur de dimensions  $32 \times 8 \times 8$ .

On peut donc ajouter la couche suivante:

---


$$\text{Conv2D } (F = 1, K = 1, P = 0, S = 1) + \text{Sigmoid}$$


---

Ce qui nous donne un tenseur de dimensions  $1 \times 8 \times 8$ , donc les valeurs sont comprises entre 0 et 1. On peut donc entraîner le réseau pour qu'il optimise les paramètres de cette dernière couche.

### Section B

Le problème avec ce genre de segmentation c'est que le réseau ne permet pas d'exploiter un contexte en dehors des pixels compris dans chaque case de la grille  $8 \times 8$ . Par exemple, dans cette photo, la case qui n'est pas correctement segmentée contient essentiellement un fond blanc, et ne contient donc pas de caractéristiques discriminantes pour identifier un chien. Si le réseau était en mesure d'utiliser les informations des cases voisines, il serait possible de segmenter correctement la case avec fond blanc, car le contexte serait disponible. Le champ réceptif de chaque neurone de sortie est de  $26 \times 26$ , tandis qu'un élément de la grille est de  $16 \times 16$ . Ainsi, très peu d'informations des cases voisines sont utilisées pour classifier une case.