 <p>INSEA</p>	<p>3<sup>ème</sup> année DSI (S5) &amp; M2 M2SI</p> <p>Machine &amp; Deep Learning</p> <p>Par : M. N. SAIDI / I. AL SAIDI</p> <p><b>Examen</b></p> <p>Durée : 1h</p>	<p>Année universitaire : 2022/2023</p>
------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------

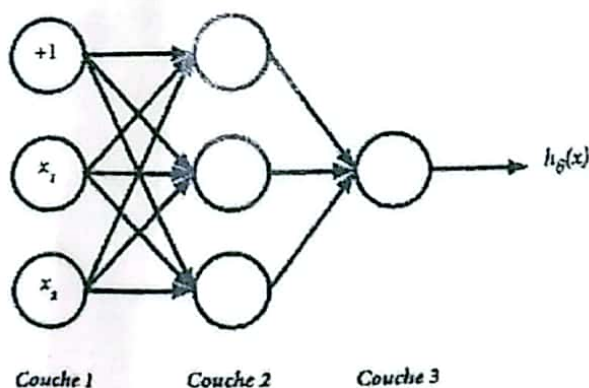
**Exercice 1 (8 pts) :**

1. Dessiner la frontière de décision, en précisant la position des classes  $y = 1$  et  $y = 0$ , trouvée par la régression logistique utilisant :

$$h_{\theta}(x) = h(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$$

Avec :  $\theta_0 = -8$ ,  $\theta_1 = 0$ ,  $\theta_2 = 0$ ,  $\theta_3 = 2$ ,  $\theta_4 = 2$

2. Expliquez l'intérêt de normaliser les données d'apprentissage ?
3. Donner un réseau de neurones qui représente la formule suivante :  
 $Y = x_1 \text{ XOR } x_2 \text{ and } x_1 \text{ XNOR } x_3$
4. Soit le réseau de neurones suivant :



Donner les formules (en les justifiant) correspondantes à la propagation en arrière

**Exercice 3 (12 pts) :**

Parite1

Soit le tenseur A en entrée de taille  $4 \times 4 \times 3$ , et une banque de deux filtres convolutionnels F 1 et F 2, de  $2 \times 2$ .

$$A(:,:,1)$$

0	2	0	3
2	1	-3	0
-1	0	0	2
-3	0	2	-2

$$A(:,:,2)$$

3	0	1	3
0	2	0	0
0	-2	-1	-1
1	0	0	-2

$$A(:,:,3)$$

1	0	-2	3
0	-1	0	0
0	0	-2	1
1	0	-4	0

$$F_1(:,:,1)$$

3	1
1	0

$$F_1(:,:,2)$$

1	-1
0	0

$$F_1(:,:,3)$$

1	0
0	1

$$F_2(:,:,1)$$

0	1
-1	2

$$F_2(:,:,2)$$

-1	0
1	0

$$F_2(:,:,3)$$

0	1
1	-3

a) Quelle sera la taille du tenseur de sortie  $B$ , correspondant au feature map d'activation (donc avant la non-linéarité) ?

b) Calculez la fibre  $B(1,1, :)$  du tenseur de sortie.

c) Calculez la fibre  $B(2,2, :)$  du tenseur de sortie.

### Partie2 :

- 1) Quels avantages apporte la convolution par rapport à des couches fully-connected ? Quelle est sa limite principale ?
- 2) Pour les convolutions, si on veut conserver en sortie les mêmes dimensions spatiales qu'en entrée. Quelles valeurs ou types de padding et de stride va-t-on choisir ?
- 3) Est-ce que la taille du filtre de convolution n'avait pas d'impact sur le nombre de ses applications dans l'image lorsqu'on utilise une marge de type de padding qui conserve la même taille de sortie que l'entrée. Expliquez pourquoi.
- 4) Pour le max pooling, on veut réduire les dimensions spatiales d'un facteur 2. Quelles valeurs de padding et de stride va-t-on choisir ?

Type de couche	Sortie	Taille du filtre/stride
Données d'entrée	224, 224, 3	
Conv1	32 convolutions, suivie de ReLU	5×5 / S=2
Pool1	max-pooling (reduction de dim spatiales 2)	2*2
Conv2	64 convolutions, suivie de ReLU	5*5/S=1
Pool2	max-pooling (reduction de dim spatiales 2)	2*2
Conv3	128 convolutions, suivie de ReLU	7*7 / S=1
Conv4	128 convolutions, suivie de ReLU	9*9 / S=1
Pool3	max-pooling (reduction de dim spatiales 2)	2*2

Fc4	fully-connected, 1000 neurones en sortie, suivie de ReLU	
Fc5	fully-connected, 10 neurones en sortie, suivie de softmax	

- 5) Prenant le modèle présenté dans le tableau, est-ce que le modèle est correct ? Si non, quelle est le problème ? Et comment l'en rectifier en gardant les mêmes hyper-paramètres de chaque couche (Conv, Pool) ?
- 6) Pour chaque couche, indiquer le nombre de paramètres et le nombre de neurones à apprendre ? Quel est donc le nombre total de neurones et paramètres à apprendre pour ce modèle ?
- 7) Donner l'utilité, les avantages, et les différents types d'apprentissage par transfert ?
- 8) Explique le concept rappels ou « callbacks » en anglais, donner les types les plus utilisés et expliquer chacun ? Donner la fonction avec ces composantes qui vous permettent de créer un rappel de point de contrôle
- 9) Expliquez ce qu'il fait le code suivant. Quels est la différence entre les deux et l'avantage de chacun (code 1 et 2)

#### # Code1

```
from tensorflow.keras import layers
from tensorflow.keras.layers.experimental import preprocessing
from tensorflow.keras.models import Sequential
augmentation_des_donnees = Sequential([
    preprocessing.RandomFlip("horizontal"),
    preprocessing.RandomRotation(0.2),
    preprocessing.RandomHeight(0.2),
    preprocessing.RandomWidth(0.2),
    preprocessing.RandomZoom(0.2),
], name="augmentation_des_donnees")
```

#### #Code2

```
augmentation_des_donnees = ImageDataGenerator(rescale=1/255.,
rotation_range = 20,
zoom_range =0.2,
width_shift_range =0.2,
height_shift_range=0.2,
horizontal_flip =True )
```