

Exercice I. 10 pts

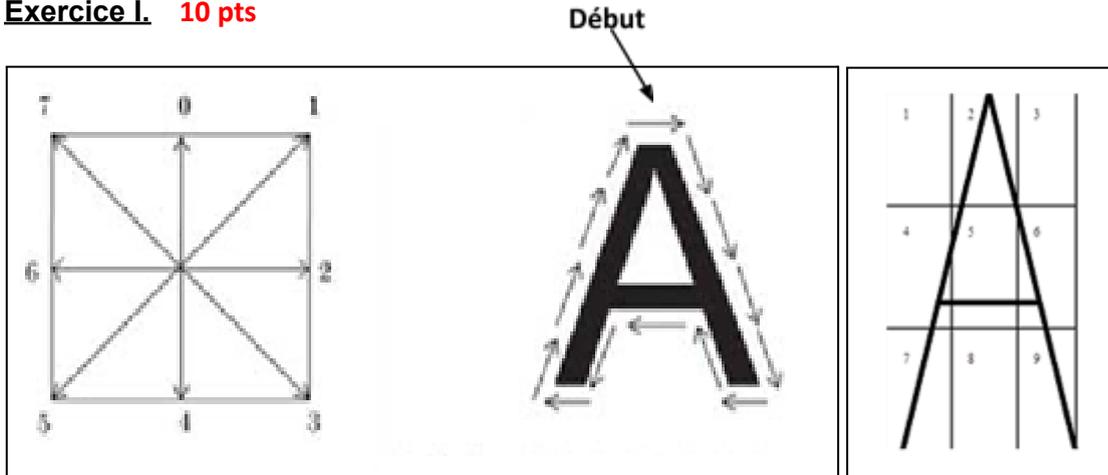


Figure 1.

Figure. 2

1.Représenter le vecteur de caractéristique de la lettre A de la figure 1 en utilisant le codage de Freeman selon les 8 directions. **1.5pt**

Le codage de Freeman est:.....2 3333676561111.....

2.En prenant en compte la décomposition de la lettre A de la figure 2 en 9 zones, proposer une représentation statistique et une autre structurale de cette lettre.**2pt**

Sur chaque zone, on extrait le pourcentage de pixels noirs. L'ensemble des 9 valeurs forme le vecteur de caractéristiques.

La forme est découpée en segments s1, s2 et s3. L'ensemble de ces segments est agencé sous forme de graphe : s1 est lié à s2, s3 coupe s1 et s2,....

3.La figure 3 représente une chaîne de reconnaissance d'une image de caractère.

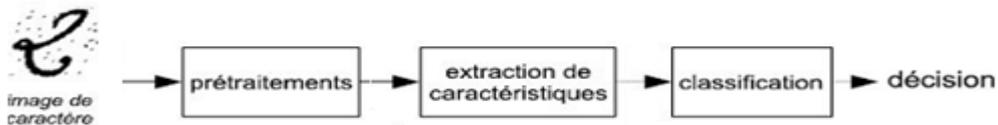


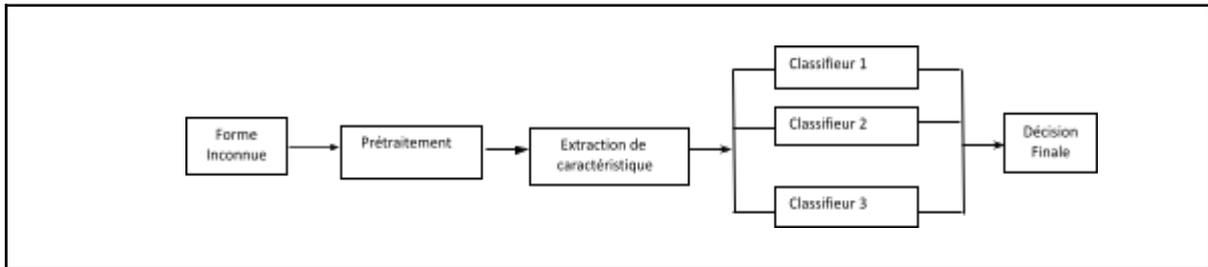
Figure. 3

Citer 4 exemples de prétraitements possibles à entreprendre pour éliminer le bruit présent dans l'image de caractère de la figure 3? **2pt**

-1.Binarisation
-
- 2.Lissage.....
-
- 3.Redressement.....
-

4. Normalisation.....

4. On souhaite étendre le système de reconnaissance de formes (RdF) illustré dans la figure 3 à un système de RdF multi-classificateur (3 classificateurs), qui se base sur une architecture en parallèle. Apportez les modifications nécessaires au schéma de la figure 3 afin de prendre en considération cette extension. **2pt**



5. Le tableau ci-dessous représente les réponses de 3 classificateurs d'un système multi classificateurs combiné selon une architecture en parallèle, pour résoudre un problème multi-classification (3 classes: C1, C2, C3). Complétez ce tableau. **0.25 x 6= 1.5pt**

	Classifieur 1	Classifieur 2	Classifieur 3	Min	Moyenne
C₁	0.1	0.4	0.2	0.1 0.25pt	0.23 0.25pt
C₂	0.0	0.7	0.9	0.0 0.25pt	0.53 0.25pt
C₃	1.0	0.5	0.5	0.5 0.25pt	0.66 0.25pt

Quelle est la décision finale obtenue en utilisant les règles de combinaison Min et Moyenne? **1pt**

La décision finale obtenue est la classe **C₃**.

Exercice II. Soit le code python suivant: **3 pts**

```

from keras.models import Sequential
from keras.layers import Dense, Activation
from keras.optimizers import SGD

model = Sequential()
model.add(Dense(128, activation='relu', input_dim=3072))
model.add(Dense(64, activation='relu'))
model.add(Dense(4, activation='softmax'))
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(optimizer=sgd, loss='categorical_crossentropy', metrics=['accuracy'])
  
```

1. Ce code python implémente un modèle de réseaux de neurones artificiel. De quel type de RNA s'agit-il? **Perceptron Multicouches(PMC).....0.75pt**

2. Remplissez le tableau ci-dessous en écrivant la signification des valeurs qui y sont mentionnées. 1.25

3072	Nombre de neurones de la couche d'entrée du PMC 0.25pt
128	Nombre de neurones de la couche la première couche cachée du PMC 0.25pt
64	Nombre de neurones de la couche la deuxième couche cachée du PMC 0.25pt
4	Nombre de neurones de la couche de sortie du PMC 0.25pt
0.01	Le taux d'apprentissage 0.25pt

3. Pourquoi le choix de la valeur du paramètre lr doit être choisi judicieusement? **1pt**

Un taux d'apprentissage trop élevé implique une divergence de l'ensemble de données (solution optimale introuvable).

Un taux d'apprentissage trop faible implique une optimisation plus longue à cause des petits pas vers les minima.

Exercice III. Soit le code python suivant: **7 pts**

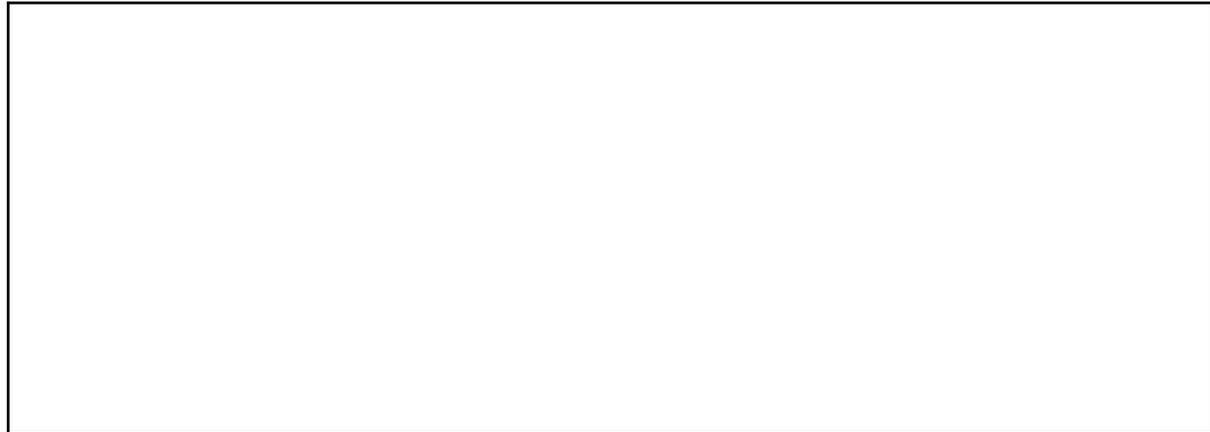
```

model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)))
model.add(Conv2D(32, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dense(4, activation='softmax'))
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', metrics=['accuracy'], optimizer=sgd)

```

1. D'après ce code, de quel type de RNA s'agit-il? **Réseau de neurones par convolution(Convnet,CNN).....1pt.....**

2. Proposer un schéma qui illustre son architecture. **2pt**



3. Soit un exemple d'apprentissage qui appartient à la classe 1 et dont le vecteur encodé en one-hot est $[1,0,0,0]$. En supposant que la sortie du RNA (raw output) génère le vecteur $[3.2, 1.3, 0.2, 0.8]$, on vous demande de convertir cette sortie en probabilité. **0.25 x 4= 1pt**

$$e^{3.2} = 24.53/24.53 + 3.66 + 1.22 + 2.22 = 24.53/31.63 = 0.775$$

$$e^{1.3} = 3.66/24.53 + 3.66 + 1.22 + 2.22 = 3.66/31.63 = 0.116$$

$$e^{0.2} = 1.22/24.53 + 3.66 + 1.22 + 2.22 = 1.22/31.63 = 0.039$$

$$e^{0.8} = 2.22/24.53 + 3.66 + 1.22 + 2.22 = 2.22/31.63 = 0.070$$

Le vecteur de probabilité est $[0.775, 0.116, 0.039, 0.07]$.

4. Appliquer la formule mathématique ci-dessous pour calculer la perte d'entropie croisée L (cross entropy loss) de l'exemple d'apprentissage précédent. **1pt**

$$L = - \sum_{i=1}^n t_i \log_2(p_i)$$

t_i : étiquette binaire de vérité de la classe i .

p_i : vecteur de probabilité.

$$L = -(1 \times \log(0.775) + 0 \times \log(0.116) + 0 \times \log(0.039) + 0 \times \log(0.07)) = 0.255$$

5. Après une mise à jour des poids, nous obtenons de nouvelles sorties brutes pour le même exemple d'apprentissage, le vecteur représentant les valeurs de probabilité devient $[0.9, 0.05, 0.03, 0.02]$. Calculer la nouvelle valeur de perte L . **1pt**

$$L = -(1 \times \log(0.9) + 0 \times \log(0.05) + 0 \times \log(0.03) + 0 \times \log(0.02)) = 0.105$$

Que remarquez-vous ? **1pt**

Les valeurs de probabilité sont plus proches de la vraie distribution (classe 1).

La nouvelle valeur de perte L est inférieure à la perte précédente, ce qui indique que le RNA est en train d'apprendre.