

UE 456 – Traitement de l'image et intelligence artificielle

Travaux pratiques n°2

Consignes à respecter :

- Indiquez vos noms / prénoms en haut de chaque fichier d'extension **ipynb** (en commentaire avec #).
- Dix minutes avant la fin du TP, envoyez une archive **ZIP** contenant vos **programmes** (sans images) et votre **compte-rendu** (PDF) sur la plateforme e-Campus (<https://ecampus.paris-saclay.fr/>). Votre compte-rendu doit détailler votre travail et les problèmes rencontrés en justifiant vos solutions. Il est inutile de remettre le code de vos programmes à l'intérieur. Pensez à illustrer les résultats obtenus par des images.
- Le barème est présent uniquement à titre indicatif et est susceptible d'évoluer.
- Le travail est **individuel**. Le fait de copier du texte ou du code provenant d'internet ou bien d'un collègue se voit tout de suite et entraînera de sévères sanctions pour toutes les parties impliquées (code ou compte-rendu).
- Plus généralement, toute consigne non respectée sera pénalisée.

Préambule

Commencez par créer un nouveau répertoire (par exemple UE_456/TP2) sur *Google Drive*. Ensuite, déposez les fichiers du TP dans ce répertoire (fichiers *.ipynb). Quelques références qui vous seront utiles pour ce TP :

- Documentation NumPy : <https://docs.scipy.org/doc/numpy/user/index.html>.
- Documentation Matplotlib : <http://matplotlib.org/>.
- Documentation OpenCV : <https://opencv24-python-tutorials.readthedocs.io/en/latest/>.
- Documentation scikit-learn : <http://scikit-learn.org/stable/index.html>.
- Documentation Keras : <https://keras.io/>.

Exercice 1 – Classification par SVM (10 points)

1. Commencez par lire, comprendre puis lancer le programme fourni. Quelle est la dimension et la nature des données étudiées ? Quelle est la taille de l'échantillon d'apprentissage et de test ? Quel est le type de modèle utilisé ? Est-il bien adapté au problème étudié ? Le volume de données disponible vous paraît-il suffisant ?
2. Multipliez par un facteur 10 le volume des données. Calculez le pourcentage d'observations bien classées avec `model.score`. Quelle est la précision et la frontière de décision obtenues sur l'ensemble d'apprentissage et de test ?
3. Passez la taille de l'échantillon d'apprentissage à 50%. Est-ce que le résultat et/ou la précision du modèle sur l'ensemble d'apprentissage et de test ont changé ? Expliquez.
4. Quel est le rôle du paramètre C ? Testez différentes valeurs à l'aide d'une boucle `for` en itérant sur `np.linspace(0.0001, 1.0, 1000)` par exemple. Comment la frontière de décision et la précision de ce modèle évoluent sur les ensembles d'apprentissage et de test ? Tracez l'évolution de la précision en fonction des valeurs prises par ce paramètre. Y a-t-il une valeur donnant de meilleurs résultats ? Cette valeur est-elle la même pour les deux ensembles ?
5. On souhaite maintenant projeter les données dans un espace de dimension supérieure. Quelle transformation proposez-vous ? Appliquez cette transformation et visualisez la projection des nouvelles données sur les variables x_1 et x_3 . Qu'observez-vous ? Recalculez le classifieur en utilisant la meilleure valeur du paramètre C trouvée à la question précédente sur les données transformées. Comparez la précision obtenue avec le modèle précédent.

Exercice 2 – Classification par CNN (10 points)

1. Rendez-vous sur la page suivante : <https://www.cs.ryerson.ca/~aharley/vis/conv/flat.html>.
2. Lisez, comprenez puis lancez le programme fourni. Quelle est la dimension et la nature des données étudiées ? Quelle est la taille de l'échantillon d'apprentissage et de test ? Le volume de données vous paraît-il suffisant ?
3. Quel est le type de réseau de neurones entraîné ? Combien le réseau comporte-t-il de couches et de couches cachées ? Quel est le nombre de paramètres à apprendre ? Comparez avec ce qui est retourné par `model.summary`. Quelle est la précision du modèle sur les données d'apprentissage et de test avec `model.evaluate` ?
4. Utilisez la fonction `get_confusion_matrix(x, model, y_true)` qui prend en paramètre les données à classer `x`, le classifieur `model` et les vraies étiquettes `y_true`, pour afficher la matrice de confusion. À quoi correspond chaque entrée de cette matrice ? Qu'observez-vous ? Comment calculer la précision du classifieur à partir de cette matrice ? Affichez des exemples parmi les données de test où le classifieur s'est trompé. Que remarquez-vous ?
5. Ajoutez au réseau précédent une couche cachée dense de taille 100 avec une fonction d'activation de type "sigmoïde" (classe `Dense`). Quel est cette fois-ci le type de réseau de neurones entraîné ? Quel est le nombre de paramètres à apprendre ? Comparez avec `model.summary`. Comparez les performances avec le réseau construit à la question précédente. Concluez.
6. Créez le réseau CNN avec les couches cachées suivantes (le reste du réseau est inchangé) :
 - (a) Couche cachée de convolution 2D de taille $64 \times 3 \times 3$ et de stride égal à $(1, 1)$ (classe `Conv2D`).
 - (b) Couche cachée de convolution 2D de taille $32 \times 3 \times 3$ et de stride égal à $(1, 1)$ (classe `Conv2D`).
 - (c) Couche cachée dense (classe `Flatten`).

Expliquez le nombre de paramètres impliqué pour chaque couche. À quoi correspond et sert la dernière couche cachée ? Mesurez la précision du nouveau réseau et comparez avec les précédents. Ensuite, insérez une couche cachée de pooling (classe `MaxPooling2D`) derrière chaque couche de convolution. Mesurez les performances et comparez.