université angers

Licence 3 Informatique
Python

Mini challenge sur la classification d'images en couleur

1 Présentation du challenge

1.1 Contexte

Pour ce projet, vous allez participer à un mini challenge de classification multi-classes sur des données réelles obtenues à partir d'images en couleur de taille 32 par 32. Vous devrez aussi fournir un court rapport qui répond aux questions posées dans le sujet et qui détaille les résultats obtenus. Ce projet est à réaliser individuellement.

Chaque image du dataset est encodée sous la forme d'un vecteur de taille 3072, au format RGB, avec 32 x 32 pixel prenant trois valeurs entre 0 et 255, correspondant à des niveaux de rouge, de vert et de bleu. Les 1024 premières entrées du vecteurs correspondent au channel rouge, les 1024 suivantes au channel vert et les 1024 dernières au channel bleu. Chaque vecteur complet de taille 3072 peut être redimensionné en un tenseur de taille (3,32,32) représentant l'image complète en couleur.

Dans ce dataset, il y a 10 types d'images différentes représentées par des classes de 0 à 9 : des avions, des voitures, des oiseaux, des chats, des daims, des chiens, des grenouilles, des chevaux, des bateaux et des camions.

Le dataset d'entraînement contient 20000 images labélisées. A partir de ces données d'entraînement, le but de ce TP est d'identifier automatiquement la classe de 10000 nouvelles images de l'ensemble de test.

1.2 Données fournies et challenge

Le challenge est organisé sur la plateforme Codalab accesible à l'url suivante : https://codalab.lisn.upsaclay.fr/competitions/22067. Il faudra au préalable créer un compte sur cette plate-forme pour pouvoir accéder aux données. Sur la plateforme de challenge Codalab, dans l'onglet Participates → Files, sont fournis les fichiers suivants :

- Un dossier zippé nommé starting k.zip qui contient deux fichiers :
  - un fichier *dataset_images_train* qui correspond aux 20000 images pour l'entraînement, sous la forme d'un dictionnaire qui a été mis au format binaire. Voici la fonction qui permet d'extraire le dictionnaire à partir de ce fichier :

```
import pickle

with open("dataset_images_train", 'rb') as fo:
    dict = pickle.load(fo, encoding='bytes')
```

  - un fichier *images_test_predictions.csv* qui donne un exemple de fichier pour les prédictions des classes des 10000 images de test. C'est ce type de fichier qu'il faudra soumettre sur le site du challenge afin d'obtenir un score de précision de votre classification.
- Un dossier zippé nommé *data_images_test* qui contient 10000 images non étiquetées, pour la phase de test du challenge.

Pour cette compétition, il s'agira d'entraîner différents modèles vus en cours en utilisant les données d'entraînement fournies. Une fois votre modèle mis au point vous pourrez l'utiliser pour classifier les images dont les données se trouvent dans le fichier *data_images_test*. Vous pourrez ensuite soumettre ce fichier de prédiction sur le site du challenge et obtenir ainsi votre score de précision pour ce problème de classification des 10000 images de la base de test.

Un affichage des meilleurs résultats obtenus par les participants est ensuite réalisé sur la plate-forme. Attention à bien valider votre modèle avant de soumettre vos prédictions sur le site car le nombre de soumissions total est limité à 50 par personne (de façon à limiter des phénomènes de sur-apprentissage des données de test).

2 Prise au main de la plate-forme de challenge CodaLab

1. Créez un compte sur la plate-forme Codalab avec un nom d'utilisateur compréhensible "prenom nom".
2. Inscrivez-vous au challenge Codalab en suivant le lien https://codalab.lisn.upsaclay.fr/competitions/22067
3. Dans l'onglet Participate puis Files téléchargez le starting kit.
4. Faites une extraction du fichier *images_test_predictions.csv* (disponible dans le starting kit), zippez ce fichier, puis faites une soumission du fichier zippé à partir de l'onglet Participate puis Submit et visualiser les résultats que vous obtenez sur le leaderboard du challenge. Il faut absolument que le fichier soit nommé exactement "images_test_predictions.csv" et soit compressé au format .zip pour que la soumission fonctionne.
   Quels résultats obtenez-vous ? (Répondre au début du rapport).
   A votre avis comment a été généré le fichier *images_test_predictions.csv* ?

3 Phase d'entraînement et de validation

3.1 Visualisations des données

Avant de proposer différents algorithmes de classification, il s'agit de visualiser les données.

1. Afficher la première image de la base d'entraînement après un redimensionnement.
2. Créez une fonction Python qui prend en entrée le numéro d'une classe, et permet d'afficher 10 images de cette classe.
3. A l'aide de la méthode TSNE de la librairie scikit-learn, réaliser une projection en deux dimensions de 3000 images de la base d'entraînement. Faites ensuite un affichage de ces points, en mettant une couleur différente pour chaque classe différente. Que constatez-vous ?

3.2 Première modélisation

Pour la première modélisation, vous utiliserez l'algorithme des k plus proches voisins vu en cours.

1. Proposez un algorithme des k plus proches voisins adapté au jeu d'entraînement du challenge.
2. Testez votre algorithme sur le jeu d'entraînement afin de trouver le meilleur nombre de voisins à prendre en compte.
3. Tracez une courbe représentant la précision du modèle en fonction du nombre de plus proches voisins. Pour cet algorithme il est conseillé de tester votre algorithme sur un fragment du jeu de données avant d'appliquer celui-ci sur le jeu complet car le traitement peut être long.
4. Appliquez votre algorithme des k plus proches voisins sur les données de test (fichier *data_images_test*) et soumettez votre prédiction sur le site du challenge. Pensez à noter les scores obtenus avec votre k plus proches voisins sur le jeu de test dans votre rapport.

3.3 Autres modélisations plus avancées

Sur le même principe que les k plus proches voisins, écrivez les algorithmes correspondants aux méthodes suivantes :
— algorithme de régression logisitique multivariée
— réseau de neurones avec des couches linéaires
— réseau de convolution (CNN)
Pour chacune de ces méthodes vous penserez à reporter les résultats et les paramètres utilisés dans votre rapport. Il sera intéressant de tester différents jeux de paramètres pour chaque méthode et de comparer les résultats obtenus.

4 Soumission du projet

Pour le 21 mars 2024 23h59 au plus tard, soumettez sur Moodle une archive contenant l'ensemble des codes que vous avez utilisés pour produire les résultats que vous avez obtenus lors du challenge ainsi qu'un document résumant les scores obtenus avec les différentes méthodes et une interprétation rapide de vos résultats.