

TP Deep Learning

Céline Jacqueton, Eloïse Giroud

12/05/2020

Contents

Introduction TP Deep Learning	2
1. D'après le contexte, à quoi correspondent les variables prédictives et cible?	2
2. Récupération d'images sur image_search_scraper.py : remarques	2
3. Proposez une méthode pour réaliser ce travail, puis affichez et interprétez le script Python dataset_splitter.py	2
4. Observez et commentez la structure de ce fichier source	3
5. Lancement de l'apprentissage sur le corpus d'apprentissage précédemment créé: remarques:	3
6. Testez ces différents modèles et interprétez les données du fichier models.csv au moyen de la technique de votre choix (par exemple avec R)	4

Introduction TP Deep Learning

L'objectif de ce TP est de prendre le contrôle d'un moteur de recherche (ici FireFoxe) et de créer un réseau neuronal pour que le système apprenne à repérer des images cohérentes de manière précise.

Nous avons choisi d'étudier le domaine "Animaux de la ferme" avec les catégories suivantes: vache charolaise, vache aubrac, cochon, cheval, chèvre

1. D'après le contexte, à quoi correspondent les variables prédictives et cible?

Les variables prédictives sont des images. Les cibles, sont des probabilités d'appartenance à des catégories. Pour chaque images, le réseau fait des calculs, puis estime un vecteur descripteur de l'image. Ensuite, une ou des dernières couches de réseau entièrement connecté permettent d'estimer à partir de ce vecteur, n probabilité correspondant à n classes/catégories d'appartenance de l'image d'entrée. La probabilité la plus élevée en sortie, donne la catégorie d'appartenance la plus probable de l'image.

2. Récupération d'images sur image_search_scraper.py : remarques

L'IA permet de produire un dossier « raw set » dans lequel on trouve 100 image de chaque catégories correspondant à notre domaine d'étude, trouvées sur le moteur de recherche FireFoxe. Ensuite nous trions manuellement ces images brutes pour enlever les images incohérentes.

En effet, nous remarquons que les images ne correspondent pas toutes à ce qu'on aurait voulu. On ne reçoit pas forcément l'animal vivant mais aussi des dessins et ce n'est pas ce qu'on souhaite apprendre à l'IA. Ce sont ces images qu'ils faut supprimer pour préciser notre demande.

Lorsque la photo comprenait une partie du corps de l'animal, plusieurs animaux ou un animal avec un homme nous avons supprimées les images pour ne pas induire l'IA en erreur.

Il est en effet important que la base d'apprentissage soit représentative (contiennent des images de toutes les catégories que l'on cherche à prédire), et qu'elle ne comporte pas d'erreurs ou d'ambiguïté (image de même catégories, classées dans deux catégories différentes par erreur).

3. Proposez une méthode pour réaliser ce travail, puis affichez et interprétez le script Python dataset_splitter.py

Il faut entrainer le système à la reconnaissance d'images avec le « training-set » pour qu'il reconnaisse précisément la demande chèvre en tant que chèvre par exemple. Pour tester qu'il a bien appris, nous utilisons le test set pour voir combien d'images il reconnaît comme étant réellement une chèvre dans x images de chèvres.

Il ne faut pas de données présentes dans deux ensembles à la fois. Il faut aussi que les données des ensembles soient représentatives (balaient l'ensemble des catégories possibles).

Le nombre de données de la base d'apprentissage doit être plus important que celui de la base de test.

Les données de la base d'apprentissage servent à mettre à jour les poids du réseau. Les données de la base de test servent à calculer une erreur de prédiction, et ainsi savoir si le réseau a suffisamment appris.

4. Observez et commentez la structure de ce fichier source

la structure du fichier `simple_cnn_menu.py` se présente ainsi :

- une partie définissant la structure du réseau de neurones convolutif (nb de couches, type de couches)
- une partie mettant en forme les différentes bases (apprentissage, test).
- une partie lançant l'apprentissage avec le réseau, et les base d'apprentissage et test
- une dernière partie qualifiant le réseau par estimation en utilisant la base de validation et mesure des erreurs.

5. Lancement de l'apprentissage sur le corpus d'apprentissage précédemment créé: remarques:

Normalement , au fur et à mesure des itérations, le taux d'erreur de prédiction doit baisser.

Voici les résultats que nous avons obtenus avec le domaine étudié:

CNN multiclass (6 catégories) avec 10 passes

- cheval : 5 / 10 (50 %)
- chevre : 3 / 14 (21 %)
- cochon : 4 / 11 (36 %)
- poule : 12 / 17 (70 %)
- vache_aubrac : 9 / 18 (50 %)
- vache_charolaise : 4 / 16 (25 %)
- Global : 37 / 86 (43 %)

CNN multiclass (6 catégories) avec 25 passes

- cheval : 5 / 10 (50 %)
- chevre : 5 / 14 (35 %)
- cochon : 4 / 11 (36 %)
- poule : 11 / 17 (64 %)
- vache_aubrac : 8 / 18 (44 %)
- vache_charolaise : 5 / 16 (31 %)
- Global : 38 / 86 (44 %)

Nous remarquons en étudiant les statistiques obtenues que le pourcentage de réussite global n'a augmenté que de 1% en faisant * de 2x * de passes. La solution a préférer est donc peut-être d'augmenter le corpus d'image de base (réponse en 6)

6. Testez ces différents modèles et interprétez les données du fichier `models.csv` au moyen de la technique de votre choix (par exemple avec R)

On fait une ACP pour déterminer quels facteurs comptent le * pour avoir le meilleur pourcentage d'apprentissage. Ensuite selon la réponse de l'ACP on peut tracer des graphiques du pourcentage de réussite en fonction de la/des variables sélectionnées .

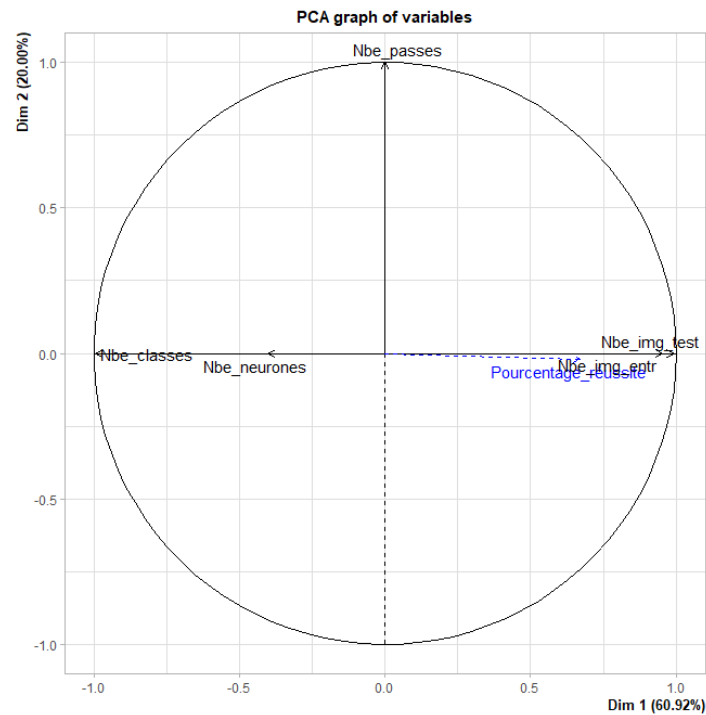


Figure 1: Graphique de corrélation des variables (noms des variables raccourcis pour plus de visibilité)

A la lecture du graphique de corrélation des variables, nous remarquons comme attendu que le pourcentage de réussite est corrélé avec le nombre d'images test et le nombre d'images d'entraînements.

Nous nous attendions à ce que le pourcentage de réussite soit également corrélé positivement au nombre de neurones. Pourtant il semble qu'ils soient corrélés de manière négative. Il se peut qu'au delà d'un nombre de neurones trop important, le système soit confronté à un surapprentissage: par exemple, Si un réseau est trop entraîné, c'est à dire qu'il arrive surgénéraliser, il est possible qu'en étant entraîné pour reconnaître des animaux de la savane, il arrive à reconnaître un chat comme un lion. Dans ce cas le pourcentage de réussite diminue lorsque le nombre de neurones augmente.

Dans ce cas, nous pensions que le pourcentages de réussite serait corrélé positivement ou négativement au nombre de passes. Or cela ne semble pas être le cas. Cela pourrait signifier que le nombre de passes nécessaire à une reconnaissance optimum doit être déterminé manuellement ("à taton") avec différents essais.