

Rapport IA sur la reconnaissance d'images

Perrine Truche, Anne Pinatel, Margot Challand

07/05/2020

Contents

1- Essais	1
1.1- <i>Domaine</i>	1
1.2- <i>Catégories</i>	1
1.3- <i>Remarques sur la constitution du corpus d'apprentissage</i>	1
1.4- <i>Remarques sur la création d'un réseau neuronal son utilisation et sa performance</i>	2
2- Etude de réseaux neuronaux à partir de graphique	3
3- Remarques sur le module	5

1- Essais

1.1- Domaine

Nous avons choisi pour nos essais d'intelligence artificielle de tester la reconnaissance des animaux.

1.2- Catégories

Nous avons fait le choix de tester les catégories suivantes : abeilles, éléphant, gazelle, giraffe, hyppopotame, léopard, lion, singe et rhinocéros.

1.3- Remarques sur la constitution du corpus d'apprentissage

La construction d'un corpus d'apprentissage d'images représentative des futures images à reconnaître se fait en plusieurs étapes. Tout d'abord, nous avons utilisé le script `image_search_scraper.py` pour télécharger d'internet 100 images de chacune des catégories que nous avons choisies. Pour cela nous avons modifié la ligne 213 pour y écrire notre propre requête : `query = ["tarte au citron", "tarte à la fraise"]`. Ces photos sont chargées dans notre fichier et dans un dossier intitulé "dataset". Nous avons ensuite passé en revue chaque photo. En effet, les photos téléchargées sont les résultats d'une recherche d'internet, ainsi toutes les photos ne sont pas forcément représentatives de la catégorie, il faut donc les supprimer manuellement. Nous avons utilisé le script `dataset_splitter.py`. Ce dernier a récupéré les photos contenues dans le fichier "raw_set" et les a réparties aléatoirement dans deux sous-fichiers. Le premier sous-fichier contient 80% des photos du fichier initial, il s'intitule "training-set" et il servira à l'apprentissage du réseau neuronal. Le

deuxième sous-fichier contient les 20% de photos restantes, il s'intitule "test_set" et servira à tester les réseau neuronal entraîné.

1.4- Remarques sur la création d'un réseau neuronal son utilisation et sa performance

Le script `simple_cnn_menu.py` est l'algorithme qui permet la création et l'apprentissage d'un nouveau réseau neuronal à partir d'un corpus d'apprentissage. On a ensuite, entre autre, la possibilité d'enregistrer le réseau construit dans un fichier, puis de l'utiliser pour déterminer la catégorie d'une image test.

Concrètement ce qu'on fait avec ce programme c'est lui apprendre à reconnaître des images (via son "système neuronal"). Il balaie les images de façon aléatoire, puis s'adapte pour reconnaître des images de catégories différentes en les triant. A la suite de cet apprentissage on lui fait faire des tests statistiques : à savoir, on lui donne de nouvelles images pour qu'il essaie à nouveau de les reconnaître. Dans l'exemple ci-dessous, nous verrons via les pourcentages combien d'images le programme parvient à bien identifier. On lui demande donc ce qu'il a reconnu !

Reconnaissance des images

Nous avons lancé un premier apprentissage à "5 passes" avec le corpus d'apprentissage pour que le réseau neuronal apprenne à différencier les images des catégories du domaine choisi. Nous avons observé que plus on augmente le nombre de passes, plus la création du réseau neuronal prends du temps, mais plus la reconnaissance est précise. Le réseau neuronal devient cependant moins apte à reconnaître d'autres images.

Le hasard correspondant à une chance sur deux soit 50%, pour que notre programme d'intelligence artificielle soit intéressant, son taux de réussite doit être supérieur à 50% et idéalement tendre vers 100%.

La figure ci-dessous est une capture d'écran montrant que le programme a reconnu 87% d'images d'abeilles. Ainsi, 87% de reconnaissance sur une dizaine de catégories est plutôt bien. En effet, plus on ferait tourner ce programme plus l'analyse serait précise.

```
0 - quitter
1 - créer et entraîner un nouveau réseau neuronal
2 - charger un réseau précédemment créé
3 - sauvegarder le réseau courant
4 - classer une image de votre disque dur
6 - classer un dossier d'images
7 - stats sur un dataset de tests
Votre choix ? : 7
Chemin d'accès pour le dossier des images (dataset/test_set) : dataset/test_set

CNN multiclass ( 10 catégories)
abeille : 43 / 49 ( 87 %)
elephant : 25 / 49 ( 51 %)
gazelle_grant : 44 / 52 ( 84 %)
gazelle_thomson : 25 / 49 ( 51 %)
giraffe : 51 / 51 ( 100 %)
hyppopotame : 4 / 50 ( 8 %)
leopard : 42 / 45 ( 93 %)
lion : 39 / 49 ( 79 %)
monkey : 37 / 50 ( 74 %)
rhinoceros : 11 / 50 ( 22 %)
Global : 321 / 494 ( 64 %)
```

Cette deuxième image reflète ce que le programme a reconnu. On remarque donc que la première image a mal été identifié tout comme l'image n°37, n°50,51,58 puis d'autres encore. Les autres images sont correctement reconnues. Cette mauvaise identification peut s'expliquer par des écritures sur l'image, des pixels un peu sombres, des couleurs pas assez marquantes. . .

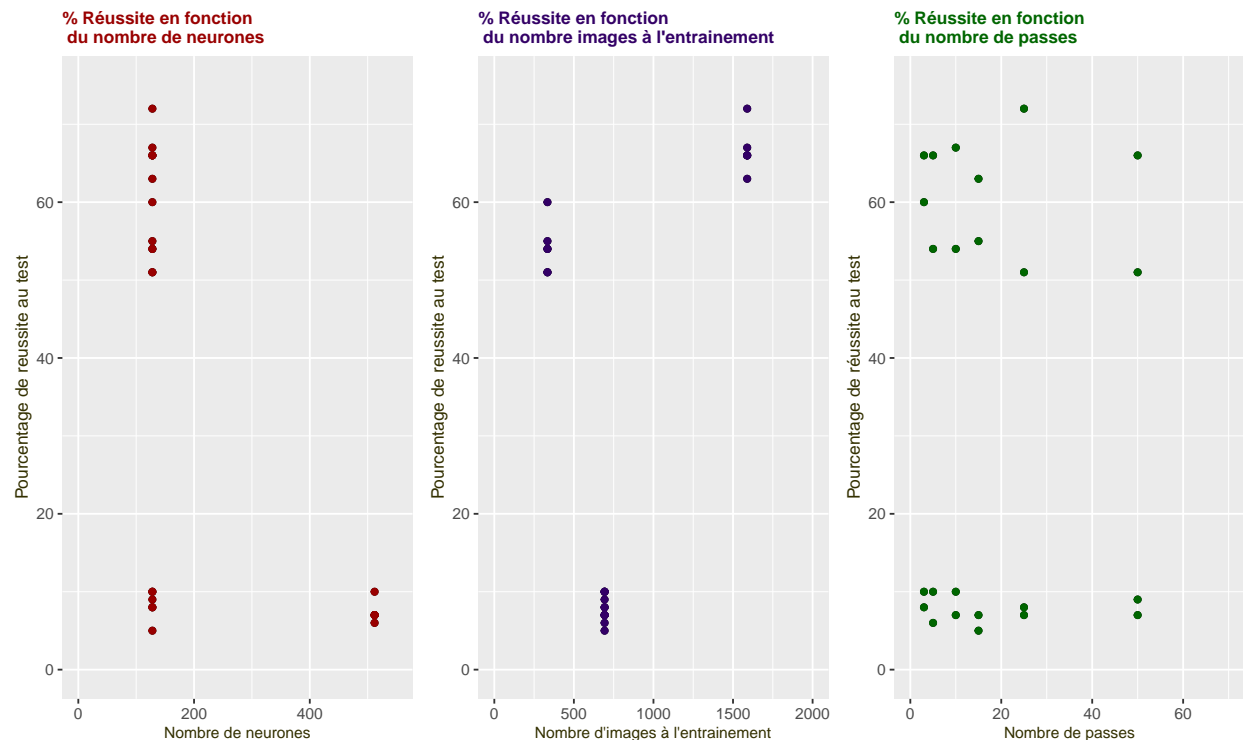
```

Chemin d'accès pour le dossier des images (dataset/test_set) : dataset/test_set
dataset/test_set\abeille\img_0.jpg giraffe
dataset/test_set\abeille\img_1.jpg abeille
dataset/test_set\abeille\img_11.jpg abeille
dataset/test_set\abeille\img_20.jpg abeille
dataset/test_set\abeille\img_21.jpg abeille
dataset/test_set\abeille\img_22.jpg abeille
dataset/test_set\abeille\img_24.jpg abeille
dataset/test_set\abeille\img_26.jpg abeille
dataset/test_set\abeille\img_27.jpg abeille
dataset/test_set\abeille\img_29.jpg abeille
dataset/test_set\abeille\img_32.jpg abeille
dataset/test_set\abeille\img_35.jpg abeille
dataset/test_set\abeille\img_36.jpg abeille
dataset/test_set\abeille\img_37.jpg giraffe
dataset/test_set\abeille\img_4.jpg abeille
dataset/test_set\abeille\img_40.jpg abeille
dataset/test_set\abeille\img_41.jpg abeille
dataset/test_set\abeille\img_43.jpg abeille
dataset/test_set\abeille\img_45.jpg abeille
dataset/test_set\abeille\img_5.jpg abeille
dataset/test_set\abeille\img_50.jpg leopard
dataset/test_set\abeille\img_51.jpg leopard
dataset/test_set\abeille\img_54.jpg abeille
dataset/test_set\abeille\img_57.jpg abeille
dataset/test_set\abeille\img_58.jpg leopard
dataset/test_set\abeille\img_60.jpg abeille
dataset/test_set\abeille\img_61.jpg giraffe
dataset/test_set\abeille\img_62.jpg abeille
dataset/test_set\abeille\img_66.jpg abeille

```

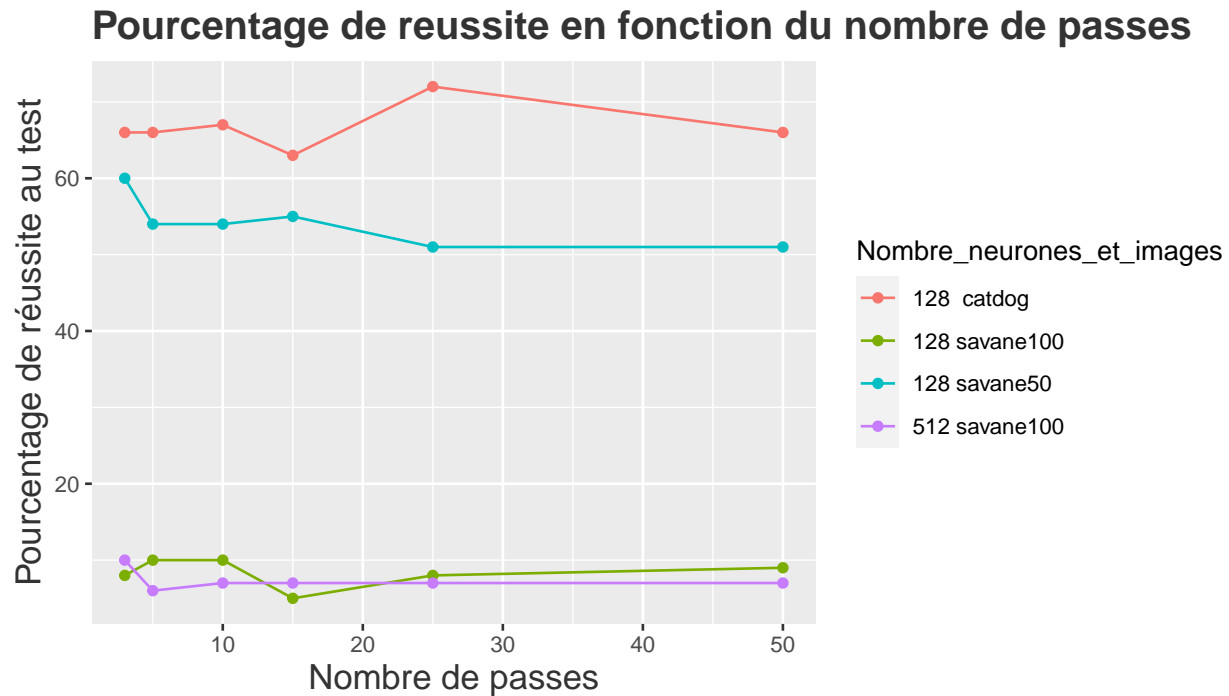
2- Etude de réseaux neuronaux à partir de graphique

Il nous a été demandé d'analyser un jeu de données sur de la reconnaissance d'images. Ce tableau de 9 variables reprend les différents facteurs qui vont nous permettre d'obtenir un pourcentage de réussite au test de reconnaissance par la machine. Un pourcentage élevé signifie que le test de reconnaissance est réussi en fonction de ce qui est cherché et ce qui est attendu. Afin d'expliquer de quoi dépend le pourcentage de réussite, nous avons décidé de tester séparément y en fonction de x, x étant chaque facteur : Nombre de neurones / Nombre d'images à l'entraînement / Nombre de passes. Si l'on prend chaque facteur individuellement, nous souhaiterions obtenir logiquement une régression linéaire positive. En effet, on peut se dire que plus le nombre de neurones est important, mieux sera la réussite. Nous obtenons donc les graphiques suivants :



Ce qu'on peut remarquer en premier lieu, c'est qu'il n'y a aucune tendance qui se dégage dans les trois graphiques. Nous pouvons donc dire que le pourcentage de réussite n'est pas le résultat d'un facteur, mais qu'il s'agit bien d'une interaction entre les trois facteurs étudiés. Il faut donc exprimer le pourcentage de réussite en fonction de ces 3 facteurs. Nous avons choisi de générer un graphique qui permettrait de montrer l'influence de 3 facteurs différents sur le pourcentage de réussite. Ces trois facteurs sont les suivants

: le nombre d'images à l'entraînement, le nombre de neurones et le nombre de passes. Pour arriver à tous les représenter sur un même graphique, nous avons choisi de concaténer les colonnes nombre_de_neurones et dataset_entrainement afin de mettre en relation les facteurs nombre d'images à l'entraînement et nombre de neurones. On observe alors sur le graphique 4 courbes représentant les 4 combinaisons possibles de ces deux facteurs : 128 catdog, 128 savane100, 128 savane50, 512 savane100. Pour chacune d'elle on observe l'évolution du pourcentage de réussite en fonction du nombre de passes.



Nous pouvons observer que les réseaux neuronaux créés à partir du jeu de données catdog contenant 1 589 images et ayant 128 neurones ont un taux de réussite qui varie entre 63% et 72%. Ces réseaux neuronaux permettent d'obtenir de meilleurs pourcentages de réussite que les réseaux neuronaux entraînés avec les jeux de donnée Savane. Cela peut s'expliquer par le fait que le nombre d'images analysées est beaucoup plus important, mais aussi par le faible nombre de catégories à distinguer. Nous pouvons donc en déduire que pour entraîner un réseau neuronal à reconnaître peu de catégories, il est plus intéressant d'utiliser un jeu de donnée contenant un nombre d'images plus important.

Le jeu de données Savane50 contient 50% des images du jeu de données Savane100. On observe que les réseaux neuronaux entraînés avec le jeu de donnée Savane50 ont un pourcentage de réussite qui varie entre 60% et 51%. Ces résultats sont bien meilleurs que ceux des réseaux neuronaux entraînés avec le jeu de donnée Savane100 (pourcentage de réussite compris entre 6% et 10%). Nous pouvons en déduire que lorsqu'on veut entraîner un réseau neuronal à reconnaître beaucoup de catégories, il est plus intéressant de l'entraîner avec un jeu de données comportant moins d'images.

Nous remarquons également que le nombre de neurones ne permet pas forcément d'obtenir de meilleurs pourcentages. En effet, les réseaux contenant le plus de neurones (512) obtiennent des taux de réussite inférieurs à 10%.

Enfin, nous pouvons observer qu'augmenter le nombre de passes pour entraîner un réseau neuronal analysant peu de catégories lui permet d'être plus performant.

3- Remarques sur le module

Les points forts du module que nous pouvons noter sont la prise en compte de la multitude de données que nous pouvons manipuler et finalement des fichiers, applications, objets (graphiques...) que nous pouvons créer. Ensuite, la distance entre les collègues et les enseignants nous force à chercher par nous même afin de nous débloquent et perdre le moins de temps possible.

On notera alors que ce module nous force à faire preuve d'autonomie et **surtout** d'entraide.

Les points faibles sont plus de l'ordre de la difficulté à comprendre le codage et la programmation, lorsqu'on est encore novice dans le domaine. En effet avoir un enseignement à distance rend toutes les choses plus compliquées. Enfin, les discussions sur plusieurs plateformes telles que le chat, la visio, les conversions... en même temps rend difficile la compréhension et le focus. Toutefois la qualité de la visiocoference était d'une grande qualité.