

Laboratoire partie 2: Entraînement d'un réseau de neurone

Dans cette deuxième partie, vous utiliserez les données collectées en partie 1 pour entraîner un réseau de neurone complètement connecté de base. Le réseau de neurone est programmé en python et nous utiliserons l'éditeur de code « visual studio code ».

Dans une première partie, nous installerons le logiciel avec les extensions nécessaires afin de faire fonctionner du python et aussi l'interpréteur de commande python pour afin d'installer les librairies nécessaires au bon fonctionnement du programme.

Par la suite, nous ferons différents tests et modification au code pour voir l'effet sur l'apprentissage du réseau de neurone.

Télécharger les données

Tout d'abord, étant donné le délai possible de téléchargement, il faut commencer par télécharger les données.

Il faut se connecter sur oneDrive en ligne sur votre compte et vous déplacer vers le dossier contenant les images enregistrées automatiquement par oneDrive. Celles-ci sont enregistré dans « images->pellicule ».

Ensuite sélectionner toutes les images à télécharger et cliquer sur « télécharger dans le haut de l'écran ».

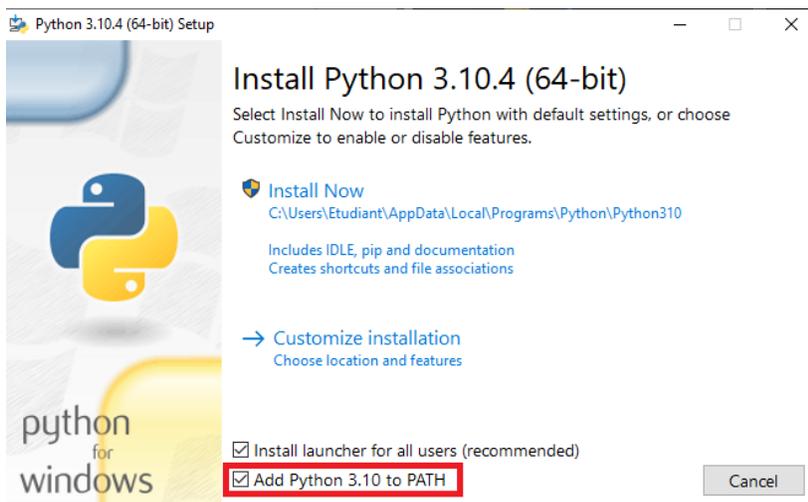
Lorsque c'est terminé, faire la même chose avec le compte du 2^e étudiant.

Installation de l'interpréteur python

Il faut d'abord installer l'interpréteur de python: <https://www.python.org/downloads/>

Cliquer sur “download python 3.10.4” Et lancer ensuite l'installation.

Pendant l'installation il faut absolument cocher “Add python 3.10 to PATH” avant de faire l'installation standard.



Installation de Visual Studio code

Visual studio code est un éditeur de programmation gratuit et efficace.

Pour installer Visual Studio code: <https://code.visualstudio.com/>

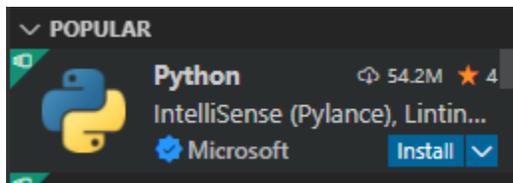
Télécharger et installer. L'installation est simple, seulement besoin d'accepter les termes du contrat et le reste des configurations par défaut sont adéquate.

Lors de la fin de l'installation, lancer le logiciel.

Installation de l'extension python

Pour utiliser le langage python, il faut ajouter une extension à l'éditeur VScode.

Pour ajouter l'extension, aller dans view->extensions. Normalement l'extension python sera visible.



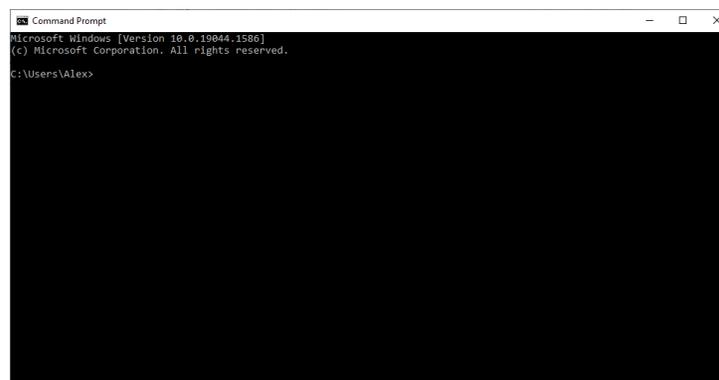
Appuyer sur Install.

Installation des librairies

Pour installer les librairies, nous utiliserons la commande « pip » qui est maintenant fonctionnelle si l'étape précédente a été bien réalisée.

Les librairies, ne sont que des fonctions et des sections de code préprogrammées qui permettent de sauver du temps et de simplifier le développement de programme. Le programme du réseau de neurones a besoin de différentes librairies.

Pour utiliser la commande « pip », il faut aller dans l'interpréteur de commande. Cliquer sur le bouton démarrer dans le coin en bas à gauche et taper « CMD » ce qui ouvrira une fenêtre comme ceci :



Pour installer les librairies : Écrire dans l'interpréteur de commande les lignes en gras ci-dessous.

Librairie numpy qui offre des fonctions mathématiques avancés.

Pip install numpy

Librairie Opencv qui permet la gestion d'images et gestion de fichiers

Pip install opencv-python

Librairie matplotlib qui permet l'affichages de graphiques. Celle-ci n'est plus utilisé dans la version finale du code mais il faut tout de même l'installer pour éviter des problèmes à l'exécution du code.

Pip install matplotlib

Librairie nnfs qui offre des fonctions utiles aux réseaux de neurones.

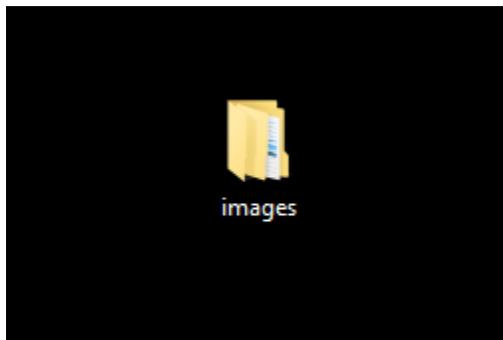
Pip install nnfs

Prétraitement des images

À cette étape, il faut que les images soient complètement téléchargées.

Il faut aussi avoir téléchargé le code sur Omnivox et placer les fichiers « `réseau_neurone_complet.py` » et « `pretraitement_image.py` » sur le bureau.

Il faut ensuite créer un dossier qui se nomme images. Lui aussi, sur le bureau.



Dans ce dossier, il faut en créer 4 autres qui se nommeront : voitures, maisons, train et test.

Name	Status	Date modified	Type
 maisons		4/7/2022 11:43 AM	File folder
 test		4/7/2022 11:45 AM	File folder
 train		4/7/2022 11:45 AM	File folder
 voitures		4/7/2022 11:43 AM	File folder

Les dossiers test et train doivent à leur tour contenir les sous-dossiers : 0 et 1.

Name	Status	Date modified	Type
0	✓	4/7/2022 11:45 AM	File folder
1	✓	4/7/2022 11:45 AM	File folder

Ensuite déplacer les images de maisons et de voitures dans les dossiers correspondant. Le programme « `pretraitement_image.py` » ira les chercher dans ces dossiers exactement.

Vous pouvez maintenant ouvrir le logiciel Visual studio code que vous venez d'installer avec les extensions nécessaires.

Dans le logiciel, ouvrir le fichier (file->open file...) « `pretraitement_image.py` » qui se trouve sur le bureau. Pour exécuter le code, dans le haut de l'interface, cliquer sur run->Start debugging->Python file. Par défaut se programme convertira les images des dossiers voitures et maisons en teintes de gris et réduira la définition à 100x100 pixels. Les images seront ensuite enregistrées dans les dossiers `images/train/0` et `images/train/1`

Normalement les dossiers devraient contenir le même nombre d'images de voiture et de bâtiments et chacun des dossiers devrait contenir au minimum 200 images. Ces images serviront à l'entraînement.

Données de validation

Pour vérifier l'apprentissage du réseau de neurone, il faut réserver une partie de la base de données. Ce sont des images qui ne seront pas utilisées dans les cycles d'apprentissage et que l'algorithme n'aura jamais traités.

Normalement, il faut réserver 10% des images pour une grande base de données. Cependant, avec une base de données plus petite, ça ne laisse pas beaucoup de données de validation. Alors, nous allons réserver 25% des images pour la validation (50 images par catégories si vous avez 200 images).

Pour ce faire, il faut simplement sélectionner les images dans les dossiers `images/train/0` et `images/train/1` et les déplacer (NE PAS COPIER, il faut déplacer) dans les dossiers correspondants `images/test/0` et `images/test/1`.

Apprentissage et tests de réseaux de neurones

Dans Visual studio code, fermer le fichier « `pretraitement_image.py` » et ouvrir le fichier « `reseau_neurone_complet.py` ».

Dans ce fichier, vous trouvez le programme en python, à certains endroits, vous allez pouvoir modifier certaines lignes de code afin de modifier le déroulement des cycles d'apprentissage et pouvoir comparer les résultats.

Voici un résumé des lignes de code que vous pourrez modifier.

- À la ligne 1046, `EPOCHS = 1`. La variable « EPOCHS » détermine le nombre de cycle d'apprentissage que fera le programme. Un cycle d'apprentissage signifie que le réseau de neurone a appris avec l'ensemble des données situées dans le dossier « train ». Initialement, cette variable est à 1.
- À partir de la ligne 1115, il y a plusieurs lignes de code semblable, ce sont les lignes qui initialise les différentes couches du réseau de neurone
 - `model.add(Layer_Dense(200, 64))`
 - Dans cette ligne de code vous pouvez modifier les chiffres 200 et 64. Le premier chiffre représente le nombre d'entrées à la couche et la deuxième ligne représente le nombre de neurones de cette couche.
 - `model.add(Activation_ReLu())`
 - Cette ligne vient juste après la précédente et vient donner la fonction d'activation.
 - Les activations peuvent être : `Activation_ReLu()`, `Activation_Sigmoid()`, `Activation_Linear()` et `Activation_Softmax()`

Initialement, le réseau ne contient qu'une seule couche d'entrée connectée directement à la couche de sortie

Exécuter le code une première fois, (run->Start debugging->Python file). Dans le terminal, Vérifier et noter le taux de succès et l'erreur moyenne (1 cycle):

Taux de succès:

Erreur moyenne:

Test du nombre de cycles

Modifier le code pour faire 5, 10, 20 et 30 cycles d'apprentissage (EPOCHS): Noter le taux de succès et l'erreur moyenne

	5 cycles	10 cycles	20 cycles	30 cycles
Taux de succès:				
Erreur moyenne:				

Que pouvez-vous conclure par rapport à l'apprentissage du réseau de neurone et le nombre de cycles d'apprentissage:

Est-ce que le taux de succès et l'erreur moyenne s'améliorent avec plus de cycles?

Est-ce qu'il est possible qu'un trop grand nombre de cycles soit nuisible? Pourquoi?

Selon vous, quel est le nombre de cycles optimal?

Test du nombre de couches

Modifier maintenant le code pour changer le nombre de couches de neurones. Pour commencer chaque couche peut avoir 200 neurones à l'exception de la couche de sortie. La fonction d'activation doit être linéaire rectifiée pour les couches internes et Softmax en sortie. Garder le nombre de cycle considéré comme optimal dans la question précédente.

Faire des tests avec 2, 5 et 10 couches au total. Noter le taux de succès et l'erreur moyenne

Nombre de couches	2	5	10
Taux de succès:			
Erreur moyenne:			

Que pouvez-vous conclure par rapport à l'apprentissage du réseau de neurone et le nombre de couches: Est-ce que plus de couches permettent d'avoir un meilleur résultat? Pourquoi?

Que remarquez-vous concernant le nombre de couches et le temps d'exécution du programme?

Quel est, selon vous, le nombre de couche optimal? Tenez compte aussi de la puissance de calcul nécessaire.

Test du nombre de neurones

Nous allons maintenant déterminer quel est le nombre de neurone optimal par couches. Conserver le nombre de couches considéré comme optimal dans la dernière question et faites des tests avec 50, 200, 1000 et 5000 neurones par couche. Noter le taux de succès et l'erreur moyenne.

Nombre de neurone par couches	50	200	1000	5000
Taux de succès:				
Erreur moyenne:				

Que pouvez-vous conclure concernant le nombre de neurones et le taux de succès du réseau dans son ensemble?

Sachant que la couche d'entrée comporte 10 000 données (100 pixels x 100 pixels). Voyez-vous une relation entre le nombre d'entrée, le nombre de neurones au total et le taux de succès du réseau de neurone? Pour déterminer le nombre total de neurones il faut additionner les neurones de chaque couche.

Selon vous, quel est le nombre de neurone optimal par couches?

Test des fonctions d'activation

Modifier les fonctions d'activation et noter les résultats. Ne pas modifier la fonction d'activation de la couche de sortie qui doit rester softmax.

Fonction d'activation	Linéaire rectifiée (par défaut)	Linéaire	Sigmoïde
Taux de succès:			
Erreur moyenne:			

Que pouvez-vous conclure concernant le choix de fonction d'activation?
