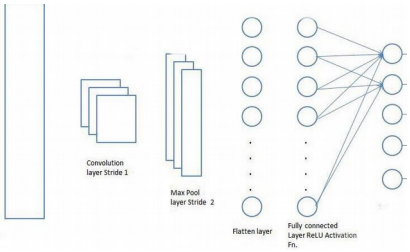
Détection des Tumeurs avec un réseau de neurones à convolution

Le But de ce projet est de pouvoir classifier et catégoriser pour prédire si une personne X présente une tumeur et pour cela on va utiliser un réseau de neurones à convolution (CNN).

La première étape, qui est l'étape primordiale, est de chercher les "Features". Je dispose de 80 000 images opensource de tumeurs pulmonaires que j'ai pu obtenir en grande partie sur "Kaggle".



C'est simple, CNN = Traitement de l'image + ANN.

Le traitement de l'image se repose sur 3 étapes :

- Convolution : Input Image x Feature Detectors = Feature Map
- Max Pooling : Max (Feature Map) = Pooled Feature Map
- Flattening : Aplatissement de la matrice "Pooled Feature Map", on prend les valeurs et on les met toutes dans un vecteur colonne ce qui va former la couche d'entrée de mon réseau de neurones

Pour ceci, on importe de la librairie Keras : Convolution2D, Maxpooling2D, Flatten. En ce qui concerne la convolution2D, on prend comme argument "Filter" qui est le nombre de mes "Features Detectors", "Kernel-size" qui est la taille de mes "Features". Ensuite, on bouge de 1 pixel qui représente la valeur de mon "Strides" et "Inputshape" dont la valeur est (64,64,3).
Quant à le Max Pooling, on prend que le "Poolsize" qui est la taille de ma petite matrice. Et pour Flatten, on a zéro argument.

Après l'ajustement des images, je passe aux couches complètement connectées. À ce niveau on choisit les fonctions d'activations et le nombre de neurones. Ce dernier est principalement calculé à travers la moyenne entre le nombre de neurones de la couche d'entrée et celui de la couche de sortie. le choix de ces paramètres est fait selon la problématique. Pour un résultat optimal, j'utilise la fonction Relu, qui est la fonction redresseur, pour les couches cachées puis la fonction sigmoïde pour la sortie. La fonction sigmoïde va nous permettre de déterminer un seuil et puisqu'on fait de la classification et on a un seul neurone de sortie, c'est l'idéal. Généralement c'est ce qu'on utilise pour la reconnaissance d'images lorsqu'on a qu'une ou deux catégories, ou alors on utilise la fonction "Soft-Max". Afin de construire les couches du réseau de neurones on importe "Dense" de Keras. Pour la compilation, on va devoir choisir l'algorithme du gradient, la fonction de coût qui dépend selon si c'est un problème de classification ou de régression.
Pour cela on utilise "Binary_crossentropy" puis on choisit aussi une métrique pour mesurer la performance de mon réseau de neurones.

La dernière étape est l'entraînement de mon réseau de neurones, j'utilise un procédé de Keras qui s'appelle l'augmentation de l'image. Elle me permet de préparer toutes les images pour éviter le surentraînement (la valeur de Training Accuracy diffère de celle du Test Accuracy). Donc au lieu de rajouter plus d'images, ce procédé va en générer de nouvelles à partir de ce qu'on a et cette méthode est "flow_fromDirectory".

Enfin, pour améliorer le modèle, on peut rajouter des couches de convolution, des couches cachées ou augmenter le nombre d'époques pour mieux ajuster les poids grâce à la rétro propagation.