

TP - Prior Neuronaux pour les images

Master ID3D - Julie Digne

14 Novembre 2024

Dans ce TP nous verrons comment utiliser les prior neuronaux pour reconstruire des images et les débruiter.

1 Librairies - logiciels

La programmation se fait en python, en utilisant les librairies suivantes

- Numpy
 - Matplotlib (pour visualiser l'évolution des fonctions de coût)
 - Pytorch (pour la partie neuronale)
 - PIL pour la lecture et sauvegarde d'image
- Un environnement conda `tp_ml.yml` se trouve sur la page du cours.

2 Prior neuronal pour la reconstruction

Dans un premier temps nous allons construire un réseau capable d'apprendre une image, soit un simple "encodeur-décodeur" avec quelques skip. En entrée nous prenons une image de bruit de 16 canaux et nous allons reconstruire une image à 3 canaux (donc une architecture pas tout à fait symétrique). L'architecture proposée est la suivante :

- Entrée : 16 canaux
- couche down 1 : 32 canaux
- couche down 2 : 64 canaux
- couche down 3 : 128 canaux
- couche down 4 : 256 canaux
- couche up 4 : 128 canaux
- couche up 3 : 64 canaux
- couche up 2 : 32 canaux
- couche up 1 : **3** canaux
- 2 skip connections aux tailles 128 et 64, les skips sont des convolutions avec des noyaux de taille 5 et 4, on concatène les résultats.

De plus :

- les activations sont des `leaky_relu` (sauf la dernière qui est une `sigmoid`)
- les couches "up" utilisent des convolutions transposées, avec des noyaux de taille 4 un stride de 2 et un padding de 1.

Implémenter le réseau et tester la reconstruction simple d'image

3 Prior neuronal pour l'inpainting

Masquer des pixels aléatoirement dans l'image de départ, et utiliser le réseau pour reconstruire l'image. Attention, la loss ne doit prendre en compte que les pixels connus.

4 Prior neuronal pour le débruitage

Bruiter l'image de départ avec un bruit gaussien, et utiliser le réseau pour reconstruire l'image.