

## Processus gaussien et Réseau de neurone 2D

On va comparer les résultats entre un réseau de neurone et un processus gaussien pour apprendre une fonction de  $R^2$  dans R.

2 réseaux de neurone :

- Le premier comporte des couches convolutionnel donc il prends en entrée l'équivalent d'une image sur  $[0,1] \times [0,1]$  et en sortie nous avons un résultat sur  $[0,1]$
- Le second est construit avec des couches denses ainsi en entrée il prends un vecteur de dimension 2 ( $x, y$ ) et en sortie un vecteur de dimension 1 ( $z$ )

2 processus gaussien

- Le premier utilise le kernel RBF
- Le second utilise le kernel matern

Résultats :

On fixe le nombre de points d'entraînement à 400

On fixe le nombre de points à prédire à 20000

Ainsi

$N_{train} = 20$  ( 20 x 20 pour  $x, y$  )

$N = 200$  ( 200x 200 pour  $x, y$  )

Réseau de neurone couche convolutionnel :

Etant donné qu'on passe des 'images' pour entraîner le modèle, on fait en sorte que la somme de toutes les points présents sur les images données soit inférieure à  $N_{train}$ . ( Dans ce model là, les points sont choisie aléatoirement pour éviter d'utiliser des points égaux)

Entrainement avec 4 images de 100 pts :

Fonction:  $f(x, y) = \sin(x + y)$   
 $mse = 0.009$   
 $temps = 5$  secondes

Fonction:  $\sin(x^*y^*2^*\pi) + \cos(\pi^*4^*y)$   
mse = 1.06  
temps = 5 secondes

Réseau de neurone couche dense:

Fonction:  $f(x,y) = \sin(x + y)$   
mse = 0.00096  
temps = 4 secondes

Fonction:  $\sin(x^*y^*2^*\pi) + \cos(\pi^*4^*y)$   
mse = 0.016  
temps = 4 secondes

Processus gaussien kernel = RBF :

Fonction:  $f(x,y) = \sin(x + y)$   
mse = 2.109 e-14  
temps = < 1 seconde

Fonction:  $\sin(x^*y^*2^*\pi) + \cos(\pi^*4^*y)$   
mse = 0.00896  
temps = < 1 seconde

Processus gaussien kernel = matern

Fonction:  $f(x,y) = \sin(x + y)$   
mse = 2.33 e-10  
temps = < 1 seconde

Fonction:  $\sin(x^*y^*2^*\pi) + \cos(\pi^*4^*y)$   
mse = 1.3e-5  
temps = < 1 seconde

On peut conclure de la même manière que pour le 1D, le processus gaussien 'fit' mieux la courbe qu' avec un réseau de neurone et est beaucoup plus rapide.

On remarque cependant que pour une fonction plus oscillante le kernel 'matern' est préférable par rapport au RBF.

Les courbes ainsi que les fonctions d'erreur sont présentes dans les fichiers .ipynb