

TP 4 : Introduction à la convolution

Jérôme Pasquet

Exercice 1 : Premier réseau convolutionnel

Nous allons nous intéresser à deux bases :

- La première est MNIST (vu lors du TP 3)
- La seconde est CIFAR10 qui est accessible via la commande : `tf.keras.datasets.cifar10.load_data()`

1. Dans le cas de CIFAR10, retrouvez les bases de train et de test ainsi que les labels associés. Affichez une image correspondante à chaque label.
2. Définir les entrées du problème et implémenter les placeholders afin de prendre en compte des images.
3. Nous allons implémenter une succession de couches de convolution contenant N_i neurones avec des kernels de taille k_i .
4. Après chaque couche de convolution nous ajouterons une couche de pooling.
5. Après les couches de convolution nous implémenterons une couche entièrement connectée. Celle-ci représentera la sortie du réseau il est donc nécessaire d'utiliser une fonction d'activation appropriée.
6. Définir et implémenter votre fonction de perte.
7. Définir et implémenter votre session et initialisez vos variables.
8. Effectuez l'apprentissage (avec du data augmentation) incluant une phase de test périodiquement. Incluez un suivi de la fonction de perte au cours du temps.
9. Répétez l'expérience en rajoutant des couches si nécessaire. Calculez le nombre de paramètres dans chacun des scénarios.

| N_1 | k_1 | N_2 | k_2 | N_3 | k_3 | $Perf$ |
|-------|-------|-------|-------|-------|-------|--------|
| 8 | 5 | 8 | 3 | - | | |
| 32 | 5 | 32 | 3 | - | | |
| 128 | 5 | 128 | 3 | - | | |
| 128 | 5 | 256 | 3 | - | | |
| 8 | 5 | 8 | 3 | 8 | 3 | |
| 32 | 5 | 64 | 3 | 64 | 3 | |
| 64 | 5 | 128 | 3 | 128 | 3 | |

10. Créer une fonction permettant de visualiser le noyaux de la première couche. Lorsque la convergence est atteinte, affichez les noyaux. Que constatez-vous ?
11. Doublez le nombre de couches de convolution en rajoutant une convolution après chaque convolution déjà présente. Le réseau sur apprend t-il le problème ? Discuter et comparer au résultat précédent. Répétez l'opération en doublant le nombre de cartes de caractéristiques.
12. Nous souhaitons modifier le réseau afin de retirer le fully connected. Nous allons donc utiliser un réseau dit : entièrement convolutionnel. Après avoir écouté les explications de votre chargé de TP sur ces types de réseau, proposer une adaptation de votre architecture afin de l'implémenter.

Devoir maison

Changez la distribution d'initialisation de vos variables. Vous utiliserez la distribution XAVIER dont vous trouverez les explications dans le papier :

Understanding the difficulty of training deep feedforward neural networks - Xavier Glorot et Yoshua Bengio