

Transfert Learning par extraction de features dans un CNN




1. Réponses aux questions

1. $7 * 7 * 512 * 4096 + 4096 * 1000 = 106 \text{ millions}$

VGG16 contient donc plus de 106 millions de paramètres.

2. La sortie est un vecteur de taille 1000 correspondant aux 1000 classes d'ImageNet.

3.

Image	Prédiction	Score de confiance
	Sweatshirt	8.10423
	Siamese cat	19.3628
	Seashore	14.3801

On voit bien qu'ImageNet n'a pas été entraîné sur des images contenant des humains car le classifieur s'est concentré sur les objets de l'image notamment les vêtements dans le premier exemple. Cependant sur des images qui ressemblent à des images sur lesquelles il a appris, le classifieur est très bon.

5. On ne peut pas directement apprendre VGG16 sur les images de 15 scenes car il n'y a pas assez d'images dans cette base de données contrairement à ImageNet qui contient plus d'1 millions d'images, cela entraînerait un sous-apprentissage.

6. Le pré-apprentissage sur ImageNet permet d'apprendre des motifs sur les images, ce qui est utile pour obtenir des vecteurs de feature représentant l'image en mettant l'accent sur les caractéristiques de l'image.

7. Les limites sont que si la nouvelles tâches que l'on souhaite faire et trop éloignée de la tâche que l'on a apprise alors les résultats ne seront pas bons. De plus, les résultats restent moins bons que si on avait pu apprendre sur une grande base de données spécifique à la tâche. Pour finir, plus on s'enfonce dans le réseau plus on spécifique à la base de données d'entraînement il faut donc faire attention à l'endroit où l'on coupe le réseau pour avoir la meilleure feature extraction possible.

8. Plus la couche à laquelle on extrait les features est profonde dans le réseau plus elle est spécifique aux données d'apprentissage. Il faut donc bien choisir où couper le réseau pour obtenir un vecteur de feature représentant suffisamment bien l'image mais en restant le plus général possible.

9. Il faut transformer les images noir et blanc en image RGB, c'est-à-dire qu'il faut passer d'une image à un canal à une image à trois canaux.

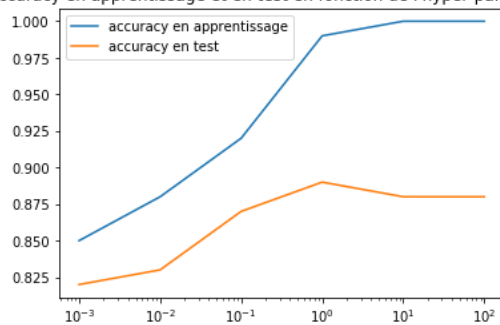
10. On obtient 0.99 d'accuracy en train et 0.89 d'accuracy en test alors qu'avec les BoW, nous obtenions 0.71 d'accuracy en train et 0.70 en test. La méthode d'extraction de features est donc meilleure.

11. Il est possible de remplacer le classifieur indépendant (SVM) par une couche fully-connected ayant pour sorti un vecteur de la taille du nombre de classe à prédire.

12. Nous avons testé plusieurs valeurs de C pour SVM

Hyper-paramètre C	Accuracy en train	Accuracy en test
100	1	0.88
30	1	0.88
10	1	0.88
1	0.99	0.89
0.1	0.92	0.87
0.01	0.88	0.83
0.001	0.85	0.82

Accuracy en apprentissage et en test en fonction de l'hyper-paramètre C

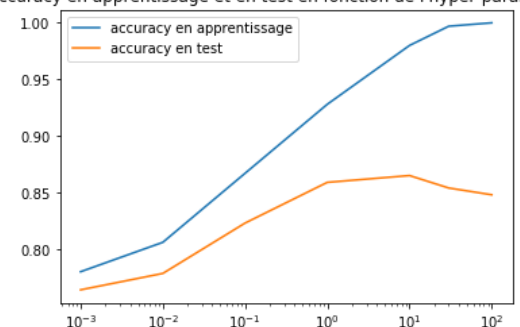


On peut voir qu'avec un C supérieur à 0.1, le modèle sur-apprend. La valeur optimale de C est 1 car l'accuracy en test est de 0.89.

Nous avons remplacé le réseau pré-entraîné VGG16 par AlexNet.

Hyper-paramètre C	Accuracy en train	Accuracy en test
100	1	0.848
30	0.997	0.854
10	0.98	0.865
1	0.928	0.859
0.1	0.867	0.823
0.01	0.806	0.7785
0.001	0.78	0.764

Accuracy en apprentissage et en test en fonction de l'hyper-paramètre C



En utilisant le modèle AlexNet nous obtenons des résultats légèrement moins bons, de plus le modèle sur-apprend un peu moins.