

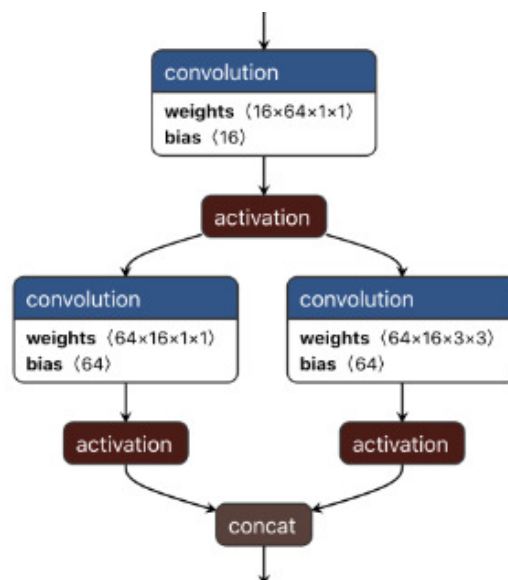
Introduction à l'apprentissage profond
Projet-INFO0702
Safia Boudra & Itheri Yahiaoui

Part 01

Construction du modèle :

Construire un CNN pour la classification multi-classes selon une architecture inspirée du modèle SqueezeNet¹. Le modèle SqueezeNet produit une accuracy égale à celle trouvé avec le modèle AlexNet mais avec 50 fois moins de paramètres grâce à la notion de Fire Module.

Un Fire Module comporte une première convolution 1*1 suivit de deux convolutions, une 1*1 et une autre à 3*3, les volumes de sortie de ces deux convolutions sont concaténés. Ceci permet de réduire considérablement le nombre de paramètres.



Fire Module

Dans ce projet, il s'agit de construire une version réduite, « *miniSqueezeNet* », qui contient 4 Fire module au lieu de 8 comme dans le modèle original.

Détails de l'architecture du *miniSqueezeNet*:

1. Input est une image RGB de taille 224*224*3
2. Les images en entrée vont être normalisées par rapport à la moyenne des plans RGB des images de ImageNet.
3. Appliquer une augmentation des données de train, avec un flip horizontal.
4. La fonction d'erreur est : loss = 'categorical-crossentropy'.
5. La méthode d'optimisation est 'Adam'
6. Taux d'apprentissage initial : lr= 0.001
7. activation='relu', strides=2, padding='same'

¹ <https://arxiv.org/abs/1602.07360>

Couche	Détails
input_1 (InputLayer) conv1 (Conv2D) maxpool1 (MaxPooling2D)	(224, 224, 3) 96 filtres de (7*7) fenetre de (3*3), strides=2
# 1er Fire Module fire1_squeeze (Conv2D) fire1_expand1 (Conv2D) fire1_expand2 (Conv2D) concatenate_1 (Concatenate)	16 filtres de (1*1) 64 filtres de (1*1) 64 filtres de (3*3) concatener les volumes output de (fire1_expand1, fire1_expand2)
# 2eme Fire Module fire2_squeeze (Conv2D) fire2_expand1 (Conv2D) fire2_expand2 (Conv2D) concatenate_2 (Concatenate)	16 filtres de (1*1) 64 filtres de (1*1) 64 filtres de (3*3) concatener les volumes output de (fire2_expand1, fire2_expand2)
# 3eme Fire Module fire3_squeeze (Conv2D) fire3_expand1 (Conv2D) fire3_expand2 (Conv2D) concatenate_3 (Concatenate)	32 filtres de (1*1) 128 filtres de (1*1) 128 filtres de (3*3) concatener les volumes output de (fire3_expand1, fire3_expand2)
# max pooling maxpool2 (MaxPooling2D)	fenetre de (3*3), strides=2
# 4eme Fire Module fire4_squeeze (Conv2D) fire4_expand1 (Conv2D) fire4_expand2 (Conv2D) concatenate_4 (Concatenate)	32 filtres de (1*1) 128 filtres de (1*1) 128 filtres de (3*3) concatener les volumes output de (fire4_expand1, fire4_expand2)
# FC dropout1 (Dropout) conv2 (Conv2D) global_average_pooling2d_1 softmax (Activation)	drop rate = 0.5 filtre de (1*1) (nbr de filtres = nb_classes) GAP Activation= softmax

La concaténation sur Keras est réalisée par la fonction Concatenate²

² https://keras.io/api/layers/merging_layers/concatenate/

Evaluation du modèle :

1. Faire une évaluation du modèle concernant :

- a) Le nombre d'epoch : [20,40,60,80,100]

- b) La taille du batch : [8, 16, 32, 64, 128]

Une fois la combinaison {epoch, batch_size} trouvée procéder à l'évaluation du taux d'apprentissage

- c) La dégradation du taux d'apprentissage :

Observez l'entraînement du modèle, s'il y a un surapprentissage, écrire une fonction `lr_decay()` à faire passer au `LearningRateScheduler`. Cette fonction applique une dégradation du taux d'apprentissage actuel d'un facteur de 0.2 à une ou des epoch(s) précise(s) pour atténuer le surapprentissage.

2. Une fois le modèle optimal trouvé après l'évaluation :

- a) Sauvegarder le modèle.

- b) Sauvegarder l'historique de l'apprentissage.

- c) Tester le modèle optimal sur l'ensemble test.

- d) Sauvegarder les valeurs des mesures de performances : accuracy, recall/precision/F1-score.

- e) Sauvegarder les graphes de loss/accuracy.

Part 02

Application le transfert d'apprentissage à partir d'un modèle CNN pré-entraîné sur ImageNet sur une nouvelle base.

Détails du transfert :

Principalement, la tâche consiste en ce qui suit :

1. Les images en entrée vont être normalisées par rapport à la moyenne des plans RGB des images de ImageNet.
2. Appliquer une augmentation des données de train, avec un flip horizontal.
3. Charger un modèle pré-entraîné sur ImageNet sans la dernière couche FC.
4. Faire un `model.summary()` avec et sans la dernière couche afin de voir la couche à redéfinir (un `globalaveragepooling`, un `dropout`, un `flatten`, etc) ceci diffère d'un modèle à l'autre.
5. Définir une nouvelle couche FC identique à l'ancienne (il faut respecter la structure originale) mais cette fois-ci initialisée aléatoirement. Le nombre de neurones en sortie est égal au nombre de classes de la base à traiter, le taux d'apprentissage doit être une valeur assez faible de 0.001.
6. Reconstruire le nouveau modèle.
7. Faire un fine-tuning sur la nouvelle FC pour un certain nombre d'epoch, afin d'apprendre des poids adaptés à la nouvelle base. Ici les autres couches sont à priori toutes gelées.
 - a) Après avoir entraîné la nouvelle FC, Geler la première couche et faire un fine-tuning sur les couches supérieures.

Il est impératif de comprendre l'architecture du modèle original afin de déterminer les blocks à geler (comme pour l'exemple présenté en TP-05, on a gelé les quatre premiers **blocks** Conv et fait un fine-tuning du **dernier block (qui comporte trois Conv)**. De même pour le modèle ResNet où il convient de geler par **block résiduel**.

Modèles CNN

Les modèles sont définis sous keras via : <https://keras.io/api/applications/>

- MobileNet : <https://arxiv.org/abs/1704.04861>
- MobileNetV2 : <https://arxiv.org/abs/1801.04381>
- NasNetMobile : <https://arxiv.org/abs/1707.07012>
- EfficientNet B0 : <https://arxiv.org/abs/1905.11946>

Evaluation du modèle : idem à page.3 (même procédure, mêmes détails)

Part 03

A l'issue de ce projet, préparer une présentation qui décrit le travail réalisé et expose vos remarques et conclusions :

Part01 : la construction du modèle, un descriptif des évaluations et les tableaux des résultats/graphes d'historiques.

Part02 : présentation de l'architecture du modèle utilisé pour le transfer learning (il est recommandé de lire et comprendre l'article original), explication du choix des couches à geler, et un descriptif des évaluations et les tableaux des résultats/graphes d'historiques.

La présentation doit être bien faite et comporte les détails de votre travail. Eviter les copier/coller des captures d'écran pour insérer les graphe/tableaux des performances.