



Sujet	Exercice n°2
Professeur	Dr. Issouf OUATTARA
Titre	Introduction au Deep Learning
Auteur	Coulibaly Cheick Ahmed

Lors de l'exécution en local sur mon PC, le notebook a rencontré des erreurs liées à l'arrêt du kernel. Après vérification, ce problème est probablement dû à des limitations matérielles (mémoire/RAM insuffisante pour l'entraînement du modèle). Afin de garantir l'exécution complète sans interruption, j'ai donc utilisé Google Colab, qui offre un environnement adapté avec davantage de ressources matérielles (CPU/GPU).

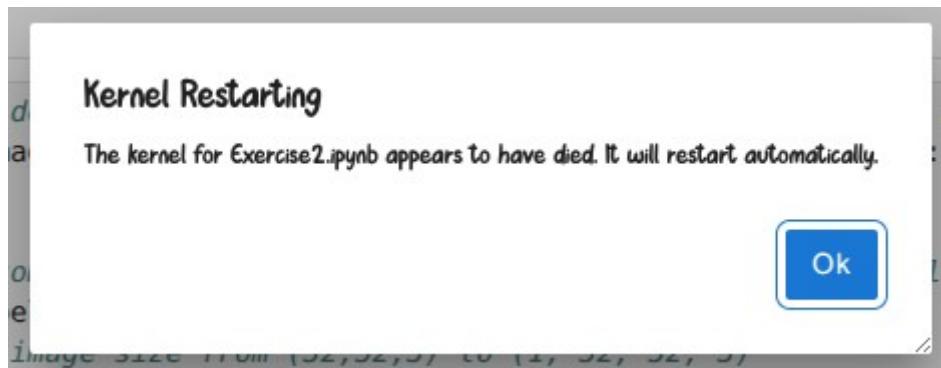


Figure 1: Erreur de kernel lors de l'exécution en local

1.1. Ajout de l'image de la courbe de fonction de coût

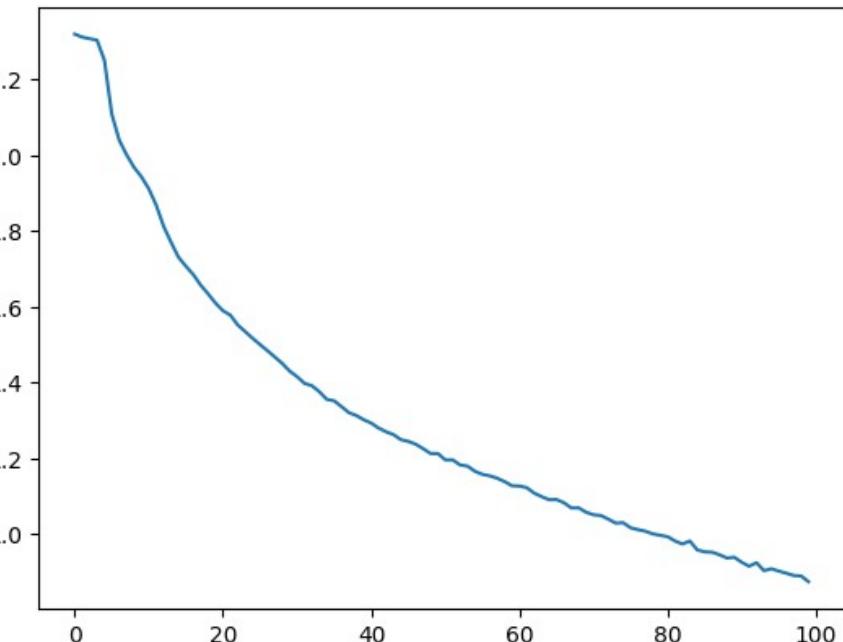


Figure 2: Courbe de la fonction de coût

1.2. Pourquoi le choix de l'architecture

Le choix de cette architecture s'explique pour plusieurs raisons :

- d'abord des paires de couches Conv2D ont été rajoutées avant les couches de pooling pour permettre au réseau d'apprendre des caractéristiques plus complexes de l'image,
 - également des couches de normalisation ont été ajoutées après chaque couche,
 - sans oublier les couches de Dropout placées après le pooling et avant la couche de sortie pour prévenir l'overfitting en désactivant un pourcentage de neurones lors de chaque étape d'entraînement,
 - la fonction d'activation ReLU est utilisée à la place de la fonction sigmoïde pour permettre une meilleure propagation du gradient , ce qui aide le réseau à apprendre plus rapidement et évite le problème de la disparition du gradient, qui est fréquent avec la sigmoïde,
 - j'ai aussi modifié la fonction train_net pour utiliser l'optimiseur Adam à la place du SGD car Adam ajuste le taux d'apprentissage pour chaque paramètre du modèle, le rendant plus performant.
- En combinant donc tout ça, le modèle a pu atteindre une performance plus élevée que le modèle de base.

1.3. Ce nouveau modèle surpassé le précédent pour divers raisons, entre autre :

- l'architecture plus profond : le modèle précédent était en effet trop simple pour capturer les nuances de images de CIFAR-10, en rajoutant donc plus de couches de convolution, le modèle est donc capable d'extraire plus de caractéristiques,
- ajouté à cela l'utilisation de la fonction d'activation ReLU,
- la batch normalisation et le dropout qui aide à prévenir le surapprentissage,
- enfin l'optimiseur Adam qui se veut plus efficace que le SGD.

En sauvegardant le modèle au format HDF5 (.h5), des WARNING ont été générés disant que ce format de fichier est considéré comme obsolète et me recommande d'utiliser plutôt le format Keras natif.
J'ai donc sauvegardé le modèle amélioré avec le format keras.

1.4 Ajout de la figure de l'évolution de la fonction de coût

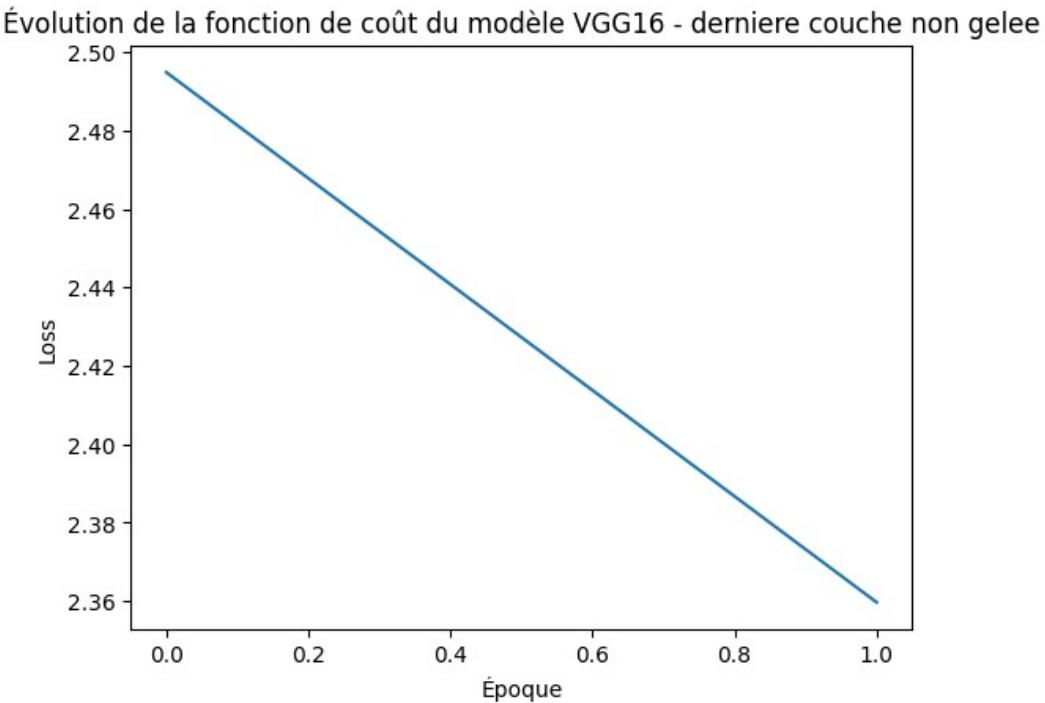


Figure 3: Courbe d'évolution - dernière couche non gelée

Il y a cependant une erreur ; en effet cette courbe est tracée après avoir entraîné le modèle avec la dernière couche non gelée et les couches gelées. Du coup, j'obtiens la même courbe, la variable 'hist' ayant servi à stocker les deux historiques. L'entraînement du modèle prenant trop de temps pour les 100 epochs, et étant en manque de temps j'ai du juste reporter avec cette courbe.

Sinon, normalement on devrait obtenir une courbe qui montre une diminution significative et continue, pas droite comme celle-ci, étant donné l'utilisation du transfert learning et du fine-tuning.