DINH Son-Tony

Etude de la performance d'un modèle de réseau de neurones basé sur InceptionV3 intégrant une méthode de régularisation L2 et d'un learning rate scheduler

I. Introduction: présentation du modèle utilisé

Le modèle utilisé a été développé sur Python par Lucas Dufour à l'aide de la bibliothèque Tensor Flow. Les phases d'entraînements ont été lancées en parallèle sur des instances de Google. Colab.

```
model = Sequential()
model.add(data_augmentation)
model.add(base_model)  # =InceptionV3
model.add(Flatten())  # Need to flatten the cnn
model.add(Dropout(0.3))
model.add(Dense(2048, activation="relu"))
model.add(Dropout(0.3))
model.add(Dense(2048, activation="relu"))
model.add(Dropout(0.3))
model.add(Dense(1024, activation="relu"))
model.add(Dropout(0.3))
model.add(Dense(512, activation="relu"))
model.add(Dropout(0.2))
model.add(Dense(512, activation="relu"))
model.add(Dropout(0.1))
model.add(Dense(256, activation="relu"))
model.add(Dense(train_generator.num_classes, activation='softmax'))
```

Figure 1: modèle de réseau de neurones utilisant le réseau pré-entraîné Inception v3

II. Présentation de l'expérience

L'expérience vise à montrer l'efficacité tout d'abord d'implémenter un système "learning rate scheduler" et ensuite d'ajouter un "RegularizerL2" à notre modèle.

A. Qu'est-ce que le "learning rate scheduler" ?

C'est une fonction qui permet lors d'une phase d'entraînement du modèle et modifier le learning rate à mesure que les époques s'écoulent. Au lieu habituellement de garder un

learning rate constant, le principe est de faire évoluer le learning rate en fonction d'une exponentielle décroissante.

Soit n le numéro de l'époque actuelle et α un réel positif. Le learning rate vérifie alors :

$$l_{n+1} = l_n \times e^{-\alpha}$$

B. Qu'est-ce que le "RegularizerL2" ?

Un "Regularizer" est un terme que l'on ajoute à la sortie d'un neurone et dont le but est d'éviter le sur-apprentissage lors de la phase d'entraînement. Sans rentrer dans les détails, il existe plusieurs méthodes de régularisation. Dans la méthode L2, le terme ajouté correspond à la somme des carrées des poids en entrée du neurone.

C. Protocole

Pour montrer l'influence de la "regularisationL2" et du "learning rate scheduler" sur notre modèle, nous allons procéder à 3 entraînements.

- Un entraînement avec "RegularizerL2"
- Un entraînement avec "Learning Rate Scheduler" avec α = 0.1
- Un entraînement utilisant les deux fonctions avec α = 0.1
- Un entraînement témoin

Les quatres modèles d'entraînement partagent des paramètres communs.

- Epoch = 20
- Default_learning_rate = 0.001
- Batch_size = 32
- optimizer = adam (SGD amélioré)

Les quatres modèlent implémentent de la data augmentation (=ajout artificiel de bruit)

Les données utilisés:

- 1073 images de Boletus
- 750 images d'Amanita (Champignons à effet secondaire)
- 75% training set
- 15% validation set
- 10% test set

III. Résultat et Interprétation

A. Résultat de la phase d'entraînement et validation

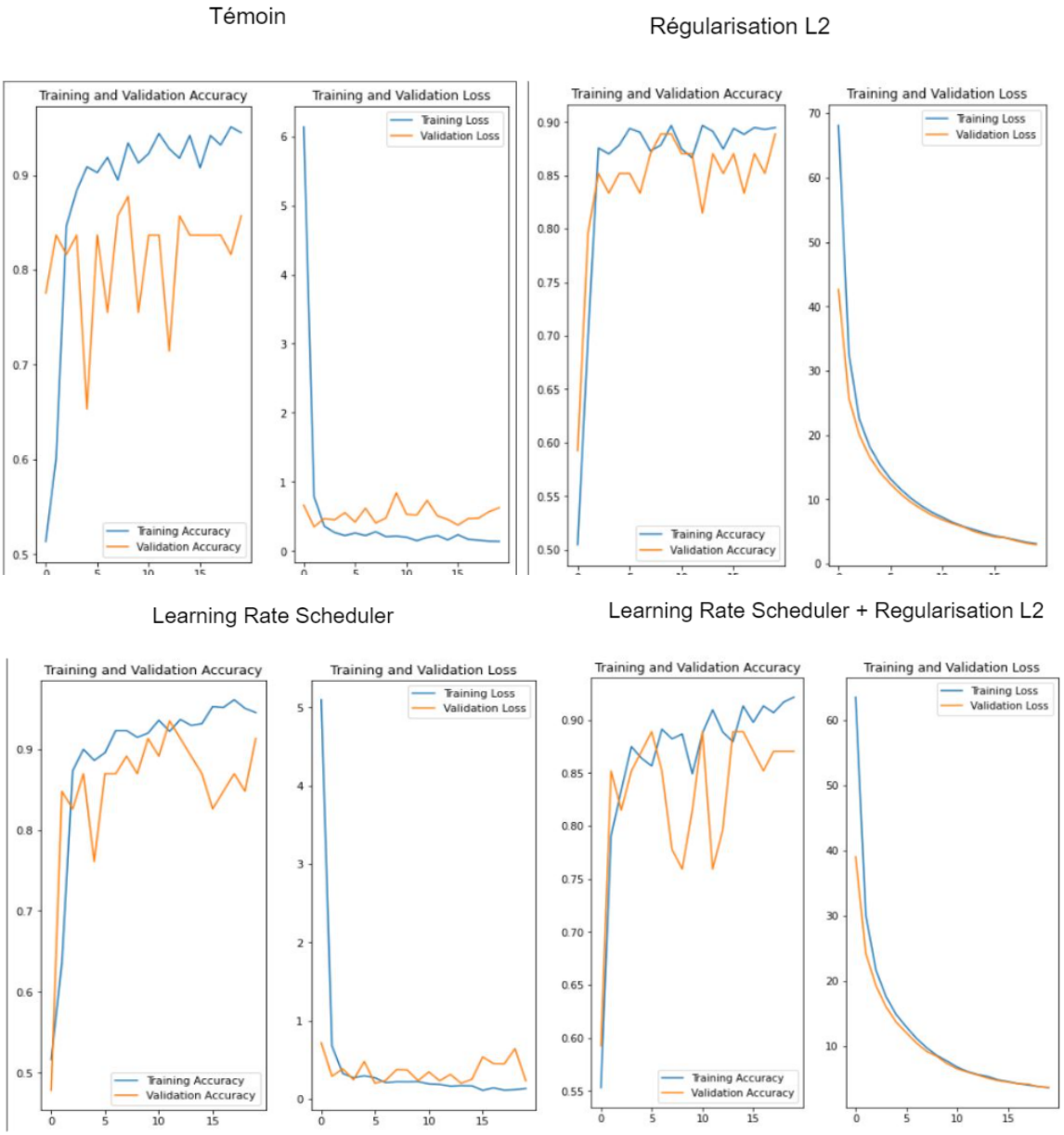

Figure 2: Graphique résultat des phases d'entraînements comprenant la précision et la perte en fonction de l'époque dans le cas de l'entraînement et de la validation

	loss	accuracy	val_loss	val_accuracy
Témoin	0.1403	0.9448	0.6272	0.8571
Lrs	0.1348	0.9451	0.2337	0.9130
ReguL2	3.1155	0.8949	2.9706	0.889
ReguL2 + Lrs	3.6003	0.9214	3.6658	0.8704

Figure 3: Tableau des résultats en fin d'entraînement pour chaque expérience

B. Observation et interprétation de la phase d'entraînement et validation

La première observation notable concerne le temps d'entraînement. En effet, ajouter la fonction de régularisation augmente le temps d'entraînement total. Le temps moyen d'une époque sans Régularisation est de 140 secondes tandis qu'avec Régularisation c'est 200 secondes. Ensuite concernant la valeur du loss dès la première époque: celle-ci est de 70 avec Regularisation et environ de 6 sans.

Enfin, on remarque qu'avec Régularisation, il y a une meilleure corrélation entre les données d'entraînement et les données de validations. Une meilleure corrélation entre ces données signifie que l'effet de surentraînement est très faible. Et donc par conséquent, les résultats du modèle en condition réel seront meilleurs, ce que nous verrons dans la partie suivante.

L'ajout du Learning Rate Scheduler semble améliorer la précision en général (figure 3). Sans Lrs, on obtient 0.85 de précision sur la validation, et 0.91 avec Lrs.

On note également que la valeur de loss est légèrement plus élevée dans le cas où il y'a régularisation que dans les cas sans. Toutefois, nous verrons par la suite qu'une valeur de loss faible ne signifie pas un modèle plus performant ou plus précis.

Des changements sont visibles dès la phase d'entraînement, cependant tant que l'on a pas procédé à la phase de test, on ne peut rien conclure quant à la performance du modèle.

C. Résultat de la phase de test

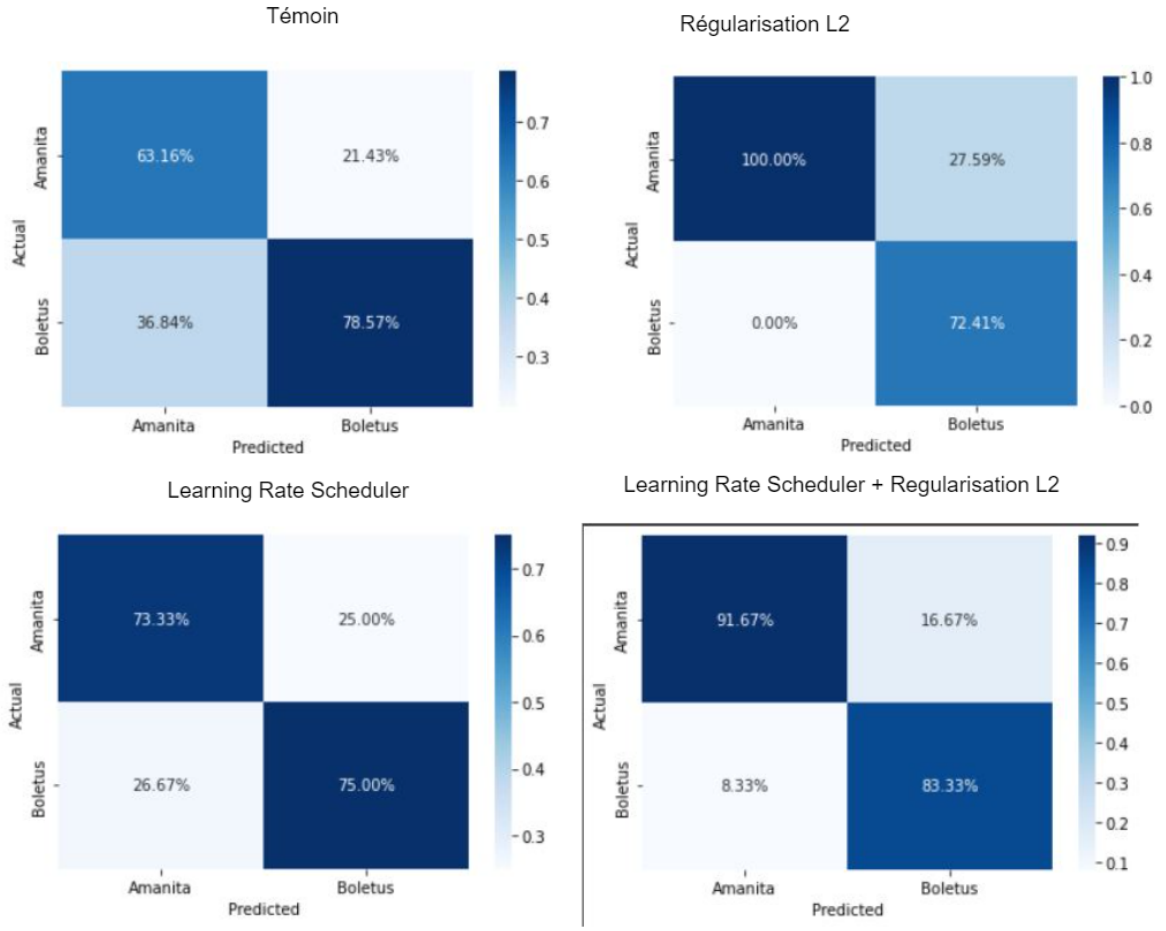

Figure 4: matrices de confusion pour chaque expérience

	Test accuracy
Témoin	70,86%
Lrs	74,16%
ReguL2	86,20%
ReguL2 + Lrs	87,34%

Figure 5: précision sur la phase de test comprenant 180 images

D. Observation et interprétation de la phase de test

Sur la figure 4 en comparant l'expérience témoin et le modèle contenant les deux fonctions, ce dernier se trompe beaucoup moins que le modèle initial. Par exemple sur une image de d'Amanita, au lieu de se tromper à 36,84%, avec le nouveau modèle, l'erreur n'est plus que de 8,33%. De même pour une image de bolée.
Ainsi, sur les mêmes échantillons de tests, le témoin réalise un précision de 70,86% alors que le modèle contenant la régularisation L2 et le "learning rate scheduler" réalise une précision de 87,34%. Ainsi l'ajout de ces deux fonctions ont permis significativement de rendre le modèle initial plus performant face à des images qu'il n'a jamais vues.

IV. Que conclure de cela ?

Après comparaison, ce qu'on peut dire c'est que le modèle intégrant la méthode de régularisation L2 et le Learning Rate Scheduler est bien plus performant que le modèle initial. L'ajout de ces fonctions ne sont pas sans contraintes puisque l'ajout de méthode de régularisation L2 rallonge de 42% le temps d'une époque d'entraînement.

De meilleurs résultats peuvent être obtenus tout d'abord en y ajoutant plus de données ou en augmentant le nombre d'époques. Ensuite, on peut également prêter plus attention aux autres paramètres de ce modèle tels que les fonctions d'activation, la méthode d'optimisation, la méthode d'initialisation des poids ou le coefficient alpha du learning rate scheduler. En procédant à du hyperparameter tuning, on peut espérer trouver les paramètres idéales pour avoir une meilleure précision.

V. Où trouver les programmes de cette étude ?

Programme Témoin :
https://colab.research.google.com/drive/1tLXU2_VQAwxY8eZBiJ_xsUmLSNciRXpR?usp=sharing

Programme RegularizerL2 :
https://colab.research.google.com/drive/1GOdK9K-p_y3HABho4U6u_PIfyeDwK0bc?usp=sharing

Programme Learning Rate Scheduler:
https://colab.research.google.com/drive/1dfDE9zD_ifnbmQiQMjOBDSwtcoDZXZ_u?usp=sharing

Programme RegularizerL2 + Learning Rate Scheduler :
https://colab.research.google.com/drive/14EiPNvKGa99RORCYS8BARRNKTWp1gTde?usp=sharing