

Classification d'images en utilisant l'API Keras

Elhouiti Chakib - Kezzoul Massili

Université de Montpellier

10 décembre 2021

1 Introduction

2 Préparation des données

3 Construction des modèles

4 Optimisation des résultats

5 Transfert learning

6 Conclusion et perspectives

Objectifs

- ▶ Chargement des données de cifar-10.
- ▶ Prétraitements des données.
- ▶ Construction et amélioration des modèles.
- ▶ Évaluation des modèles.
- ▶ Transfert Learning.



- 1 Introduction
- 2 Préparation des données**
- 3 Construction des modèles
- 4 Optimisation des résultats
- 5 Transfert learning
- 6 Conclusion et perspectives

Prétraitements

Normalisation Rangement des données dans une échelle de 0 à 1 au lieu de 0 à 255.

One hot encoding Transformation des labels d'une chaîne de caractères à des vecteur binaires.



- 1 Introduction
- 2 Préparation des données
- 3 Construction des modèles**
- 4 Optimisation des résultats
- 5 Transfert learning
- 6 Conclusion et perspectives

Première version

Structure

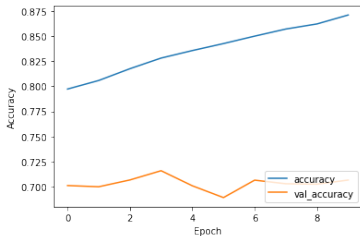
- ▶ 3 couches convolutionnelles de 32, 64 et 128 filtres.
- ▶ Chaque couche suivi d'une couche *MaxPooling*.
- ▶ Une couche *Flatten*.
- ▶ Une couche *Dense* de 256 neurones.

Hyperparamètres

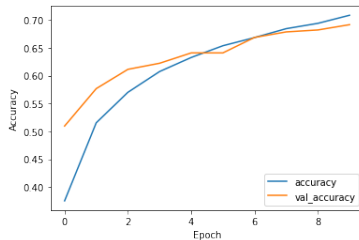
- ▶ Un optimizer *Adam*
- ▶ Learning rate de 0.005.
- ▶ Loss fonction *mean_squared_error*.
- ▶ Nombre d'epochs de 10.
- ▶ Un batch size de 128.

Évaluation et *Dropout*

Ajout d'une couche *dropout* afin de perturber le réseau lors de son entraînement.



(a) Sans Dropout



(b) Avec Dropout

Amélioration de la structure

Élargir le modèle

BatchNormalization Ajout d'une couche de BatchNormalization après chaque couche de convolution ainsi que les couches denses.

Dense Ajout d'une couche dense de 512 neurones.

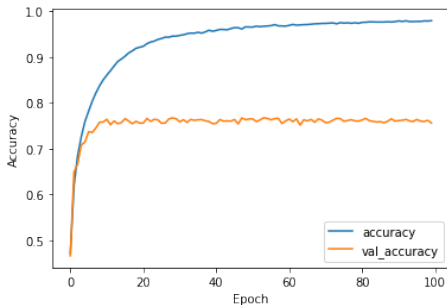


Figure – Évaluation après l'amélioration

- ① Introduction
- ② Préparation des données
- ③ Construction des modèles
- ④ Optimisation des résultats**
- ⑤ Transfert learning
- ⑥ Conclusion et perspectives

Data augmentation

- ▶ Une rotation des images ;
- ▶ Retourner l'image verticalement ou/horizontalement ;
- ▶ Décaler l'image de quelques pixels de droite à gauche ou/et de haut en bas.

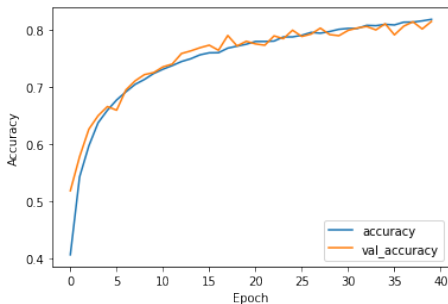


Figure – Évaluation après data augmentation

Variation des hyperparamètres

Best_params

Ça consiste à trouver les meilleurs hyperparamètres parmi : loss fonction, learning rate, optimizer, dropout, pooling.

LOSS	LR	DROPOUT	OPT	POOLING	ACC	VAL_ACC	LOSS_VALUE	VAL_LOSS
categorical_crossentropy	0.0050	0.3	adam	max	0.982699990272522	0.8134999871253967	0.05094645917415619	0.87392204999992371
mean_squared_error	0.0050	0.3	adam	max	0.9755399823188782	0.8162999749183655	0.0039016399532556534	0.028774920850982203
categorical_crossentropy	0.0010	0.3	adam	max	0.9873999953269958	0.8122000098228455	0.036502156406641006	0.9084367156028748
mean_squared_error	0.0010	0.3	adam	max	0.9668800234794617	0.8054999709129333	0.005121569149196148	0.030964171513915062
categorical_crossentropy	0.0001	0.3	adam	max	0.9850999712944031	0.6791999936103821	0.05323021486401558	1.4379479885101318
mean_squared_error	0.0001	0.3	adam	max	0.9637600183486938	0.69660001199317932	0.006020405329763889	0.0457441508769989
categorical_crossentropy	0.0050	0.5	adam	max	0.9847599864006042	0.8166999816894531	0.04526571184396744	0.840005099773407
mean_squared_error	0.0050	0.5	adam	max	0.9549199938774109	0.8058000206947327	0.006954308599233627	0.030467113479971886
categorical_crossentropy	0.0010	0.5	adam	max	0.9773200154304504	0.8014000058174133	0.06537435203790665	0.9227089881896973
mean_squared_error	0.0010	0.5	adam	max	0.9673799872398376	0.8029000163078308	0.00509100966155529	0.031911027790307999
categorical_crossentropy	0.0001	0.5	adam	max	0.9894400238990784	0.7031000256538391	0.0560577305496739	1.1684964895248413
mean_squared_error	0.0001	0.5	adam	max	0.9255399703979492	0.7019000053405762	0.011594465002417564	0.04377572610974312
categorical_crossentropy	0.0050	0.3	adam	avg	0.9678000211715698	0.809499979019165	0.09721206873655319	0.9040825366973877
mean_squared_error	0.0050	0.3	adam	avg	0.9700599808828735	0.8172000050544739	0.00472243782132864	0.02985984113812447
categorical_crossentropy	0.0010	0.3	adam	avg	0.9852799773216248	0.8251000046730042	0.04228955879807472	0.8813745975494385
mean_squared_error	0.0010	0.3	adam	avg	0.9795600175857544	0.8328899876760713	0.00328090482473135	0.026815585792064667
categorical_crossentropy	0.0001	0.3	adam	avg	0.9963799715042114	0.734499990940094	0.025248408317565918	1.082409143447876
mean_squared_error	0.0001	0.3	adam	avg	0.9676200151443481	0.7422999739646912	0.065355002824217081	0.038527511060237885
categorical_crossentropy	0.0050	0.5	adam	avg	0.9815800189971924	0.823199987411499	0.053745608776807785	0.8203244209289551
mean_squared_error	0.0050	0.5	adam	avg	0.939740002155304	0.7971000075340271	0.009217387065291405	0.032532718032589495
categorical_crossentropy	0.0010	0.5	adam	avg	0.983020007610321	0.823199987411499	0.04948342591524124	0.8062456250190735
mean_squared_error	0.0010	0.5	adam	avg	0.9724400043487549	0.8241000175476074	0.00430484856061935425	0.027826843783259392
categorical_crossentropy	0.0001	0.5	adam	avg	0.9838200211524963	0.7491000294685364	0.07062611729006631	0.9248133997781372
mean_squared_error	0.0001	0.5	adam	avg	0.9383400082588196	0.7493000030517578	0.00979839637875557	0.03647753223776817

Figure – Tableau comparant les combinaison des hyperparamètres.

Dernière évaluation

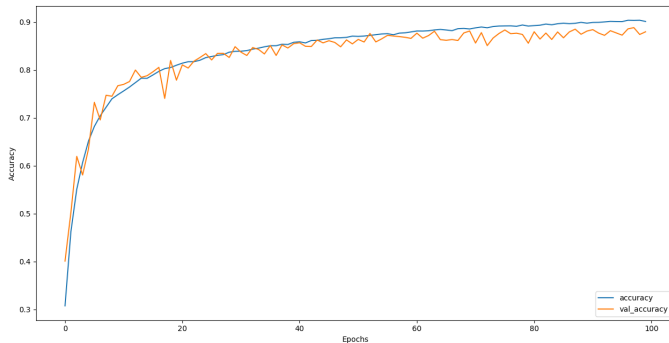


Figure – Évaluation après l'obtention des meilleurs hyperparamètres.

- ① Introduction
- ② Préparation des données
- ③ Construction des modèles
- ④ Optimisation des résultats
- ⑤ Transfert learning**
- ⑥ Conclusion et perspectives

modèles pré-entraînés

Choix de trois modèles pré-entraînés :

- ▶ MobileNetV2.
- ▶ DenseNet121.
- ▶ VGG16.

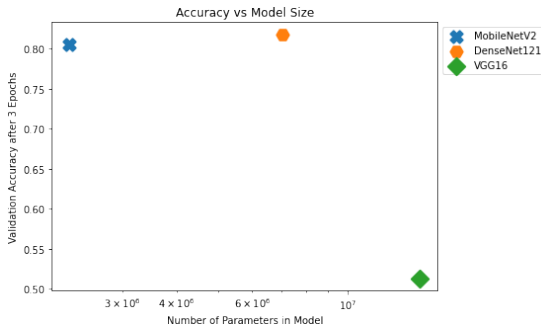


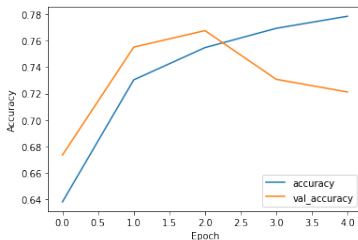
Figure – Comparaison entre modèles pré-entraînés.

Comparaison des techniques

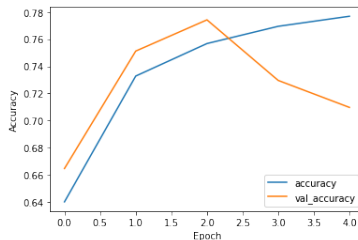
Transfert learning

Deux techniques existe pour faire du transfert learning :

- ▶ Feature Extraction.
- ▶ Fine tuning.



(a) Feature Extraction



(b) Fine tuning

Combinaison des techniques

Par contre, il est possible de combiner les deux techniques pour avoir un meilleur résultat.

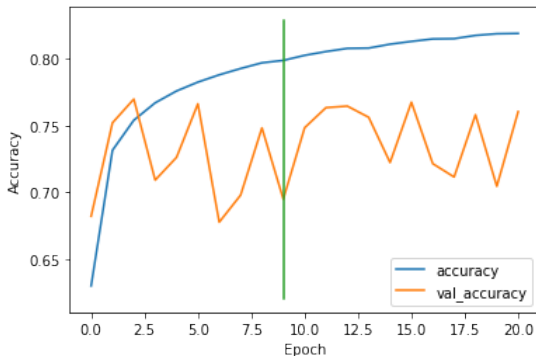


Figure – Combinaison de Feature extraction avec du fine tuning.

- ① Introduction
- ② Préparation des données
- ③ Construction des modèles
- ④ Optimisation des résultats
- ⑤ Transfert learning
- ⑥ Conclusion et perspectives**

Atous du Projet

- ▶ Data augmentation.
- ▶ Variation des hyperparamètres pour obtenir la meilleure combinaison de ces derniers.
- ▶ Transfert learning.

Perspectives

- ▶ Varier plus d'hyperparamètres et d'options de data augmentation et pouvoir aussi les varier au même temps.
- ▶ Exploiter le transfert learning avec de la data augmentation.



Merci pour votre attention.



UNIVERSITÉ
DE MONTPELLIER