



Fruits!

DEPLOIEMENT DE MODELE DANS LE CLOUD

Novembre 2022

SOMMAIRE



Rappel de la problématique



Présentation du jeu de données



Architecture cloud



Présentation de la chaîne de traitement



Conclusion



Fruits!

RAPPEL DE LA PROBLEMATIQUE



Contexte

Fruits souhaite se faire connaître en mettant à disposition du grand public une application mobile qui permettrait aux utilisateurs de prendre en photo un fruit et d'obtenir des informations sur ce fruit.



Mission

Développer dans un environnement Big Data une première chaîne de traitement des données qui comprendra le preprocessing et une étape de réduction de dimension.



Contraintes

Le volume de données va augmenter très rapidement après la livraison.

Nécessité de développer des scripts en Pyspark et utiliser le cloud AWS pour profiter d'une architecture Big Data (EC2, S3, IAM), basée sur un serveur EC2 Linux.



Fruits!

SOMMAIRE



Rappel de la problématique



Présentation du jeu de données



Architecture cloud



Présentation de la chaîne de traitement



Conclusion



Fruits!

PRESENTATION DU JEU DE DONNEES

Données sur 24 variétés de fruits

Un dossier Meta qui contient des fichiers au format texte sur les caractéristiques de chaque variété de fruits.

Des dossiers Test, Training, Validation qui contiennent pour chaque variété de fruits un sous dossier avec des images des fruits.

Nombre d'images par dossier :

Training	Test	Validation	Total
6231	3110	3114	12 455

Images de fruits de tailles et formes diverses



Fruits!

SOMMAIRE



Rappel de la problématique



Présentation du jeu de données



Architecture cloud



Présentation de la chaîne de traitement



Conclusion

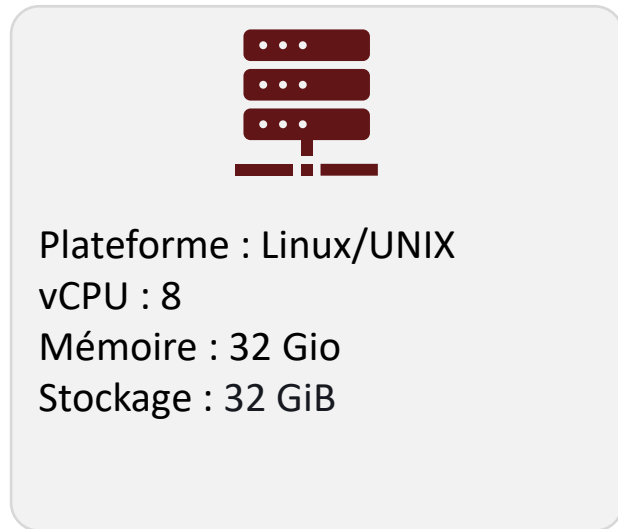


Fruits!

ARCHITECTURE CLOUD

Zone : eu-west-3c (Paris)

EC2 Instance



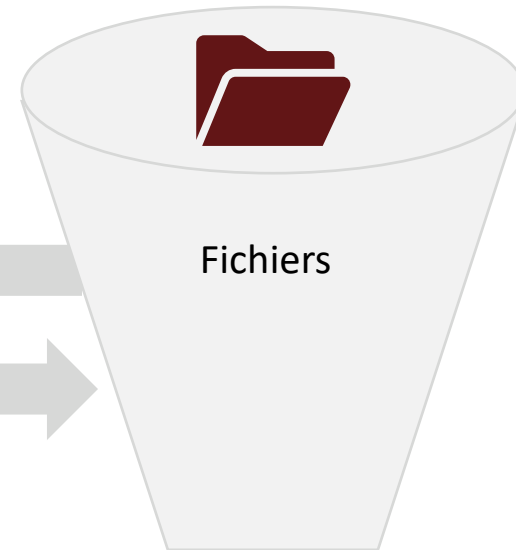
EC2
IAM Role



Read

Write

S3 Bucket



Fruits!

SOMMAIRE



Rappel de la problématique



Présentation du jeu de données



Architecture cloud



Présentation de la chaîne de traitement



Conclusion



Fruits!

PRÉSENTATION DE LA CHAÎNE DE TRAITEMENT

Modèle VGG16



Simplification du dernier layer et suppression de la couche de prédiction



Création d'une fonction de lecture, prétraitement et prédiction d'une image



Parallélisation du traitement et de la prédiction de chaque image avec Pyspark



Ecriture sur S3



Fruits!

TRANSFERT LEARNING VGG16

Input

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0

+

Output

Layer (type)	Output Shape	Param #
flatten_2 (Flatten)	(None, 25088)	0
dense_4 (Dense)	(None, 512)	12845568
dense_5 (Dense)	(None, 512)	262656

=

New Model

Total params: 27,822,912
Trainable params: 27,822,912
Non-trainable params: 0



Réduction de la
dimension des
vecteurs

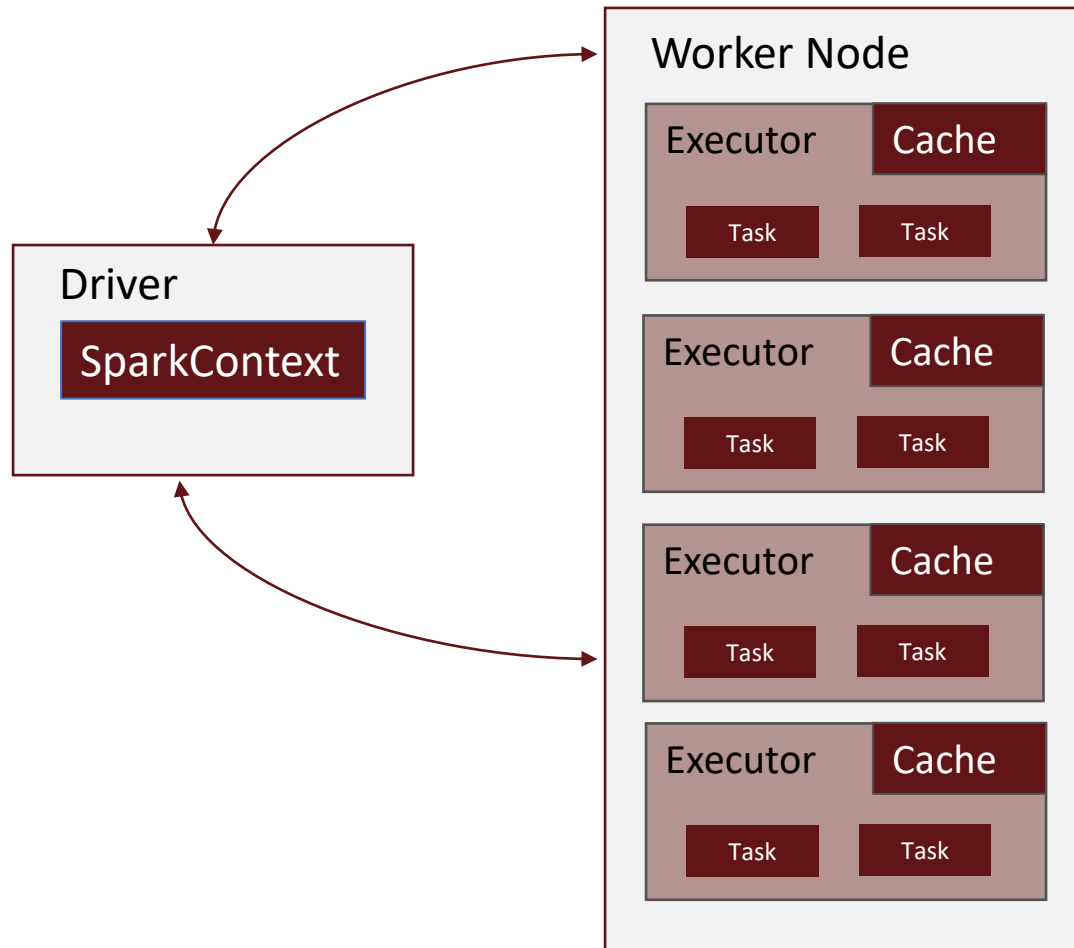
Récupération des
feature vectors

Suppression du layer
de prédiction



Fruits!

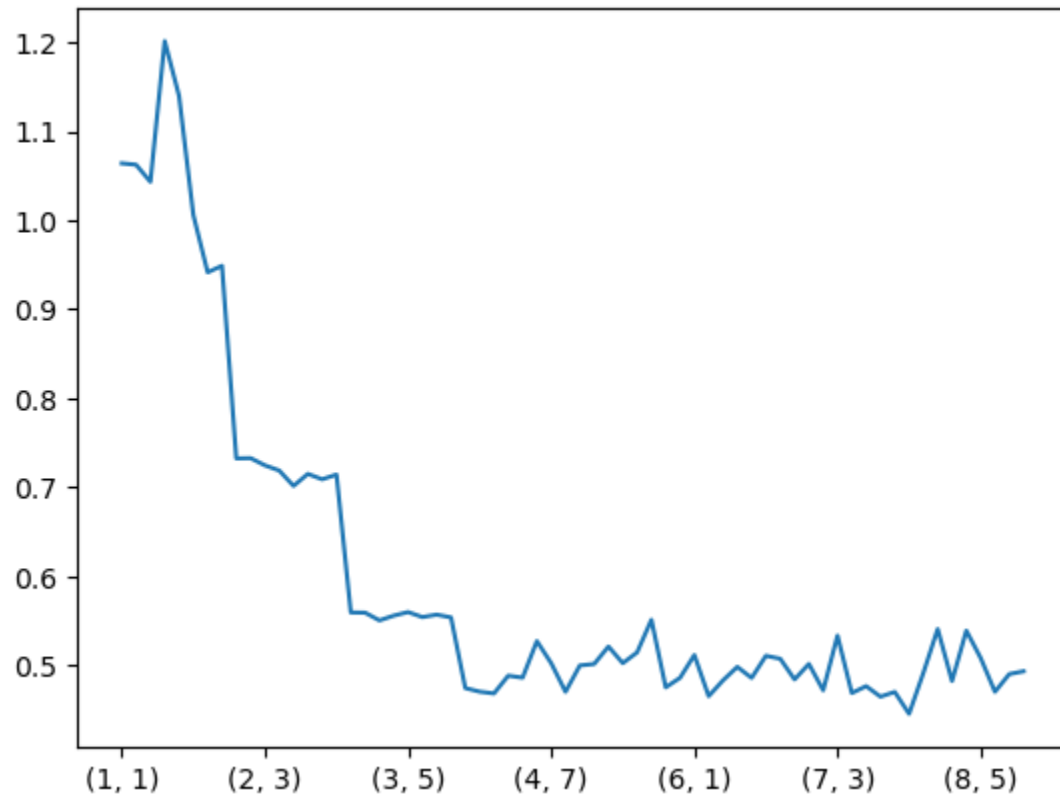
PARALLÉLISATION DU TRAITEMENT



- Utilisation en mode local : pas de Cluster manager
- Utilisation de la fonction broadcast pour charger le modèle VGG16 une seule fois sur chaque executor
- Variation des paramètres :
 - `master.Local[]`
 - `executor.instance`
 - `executor.cores`pour déterminer le nombre optimal d'executors et de coeurs par executor

PARALLÉLISATION DU TRAITEMENT

CPU time



Executors, cores

CPU time

(7, 8) 0.445396

(7, 6) 0.464660

(6, 2) 0.465255

(4, 3) 0.468283

(7, 4) 0.468711

(7, 7) 0.469994

(8, 6) 0.470150

(4, 8) 0.470159

(4, 2) 0.470293

(7, 2) 0.471859

Meilleure
combinaison

SOMMAIRE



Rappel de la problématique



Présentation du jeu de données



Architecture cloud



Présentation de la chaîne de traitement



Conclusion



Fruits!

SYNTHESE DES RESULTATS DE LA PARALLELISATION

SANS



1 000 images

Nombre de machine : 1
Nombre de CPU : 8

Total CPU TIME : 8 min 59 secondes

AVEC



1 000 images

Nombre de machine : 1
Nombre de CPU : 8

Nombre d'executors par node: 7
Nombre de Coeurs par executor : 8

Total CPU TIME : 0.44 secondes

PROJECTION



12 455 images

Nombre de machine : 2
Nombre de CPU /machine : 8

Nombre de node : 2
Nombre d'executors par node:7
Nombre de Coeurs par executor:8

Total CPU TIME estimé : 2.74 secondes



Fruits!