

Formation Machine Learning

Atelier Pratique AP-ML6

Atelier Pratique AP-ML6 : Exercice ML6.1.2

Objectif :

- Modifier le code de l'exercice **6.1.1** pour utiliser la fonction **sigmoïd** au lieu de la fonction **softmax**
- Adapter les fonctions ***Afficher_image()*** et ***Afficher_bars()***



copyright © 2021 aliase-formation.com



SOLUTION

Atelier Pratique AP-ML6 : Exercice ML6.2

Objectif :

- Faire une boucle pour déterminer les meilleurs paramètres d'apprentissage d'un modèle
- Ecrire les paramètres testés + le ratio d'erreurs dans un fichier **Liste_tests.xlsx**

Démarche : Définir les fonctions :

- *create_model(nb_filters)*
- *compiler_model(model)*
- *tester_models()*
 - : - Lance l'apprentissage (fit) sur n différents models, avec : **nb_filters** (16, 32, 64), **nb_epochs** (2, 4, 6) → 9 models.
 - Ecrit les résultats suivants dans un fichier **Liste_tests.xlsx**:
(no_modele, nb_filters, nb_epochs, train_acc, train_loss, test_acc, test_loss, resolution, nb erreurs, ratio)
- *sauvegarder_model(model, no_modele)* : sauvegarde le modele testé dans un fichier 'modele_i.backup'
- *predire (i, predictions)* : retourne la prediction de l'image N° i dans la liste des predictions
- *run_tests(nb, predictions)* : fait nb predictions sur les images du Test Set, et retourne **nb_erreurs**



Atelier Pratique AP-ML6 : Exercice ML6.3

Tester le CNN avec le dataset des **digits** :

Objectif : Classifier les images en 10 classes



```
Layer0 = layers.Conv2D(32, (3,3), activation='relu', input_shape=(8, 8, 1)) # 32 = nb filtres
Layer1 = layers.MaxPooling2D((2,2)) # (2,2) dimension de la matrice de pooling

model = models.Sequential( [Layer0, Layer1] )

LayerA = layers.Flatten()
LayerB = layers.Dense(512, activation='relu')
LayerC = layers.Dense(10, activation='softmax')

model.add( LayerA )
model.add( LayerB )
model.add( LayerC )
```



Atelier Pratique AP-ML6 : Exercice ML6.4

Tester le CNN avec le dataset [fashion_mnist](#) :

Objectif : Classifier les images en 10 classes

Créer un modèle avec 3 niveaux de convolution :

Niveau 1 : 32 filtres

Niveau 2 : 64 filtres

Niveau 3 : 64 filtres



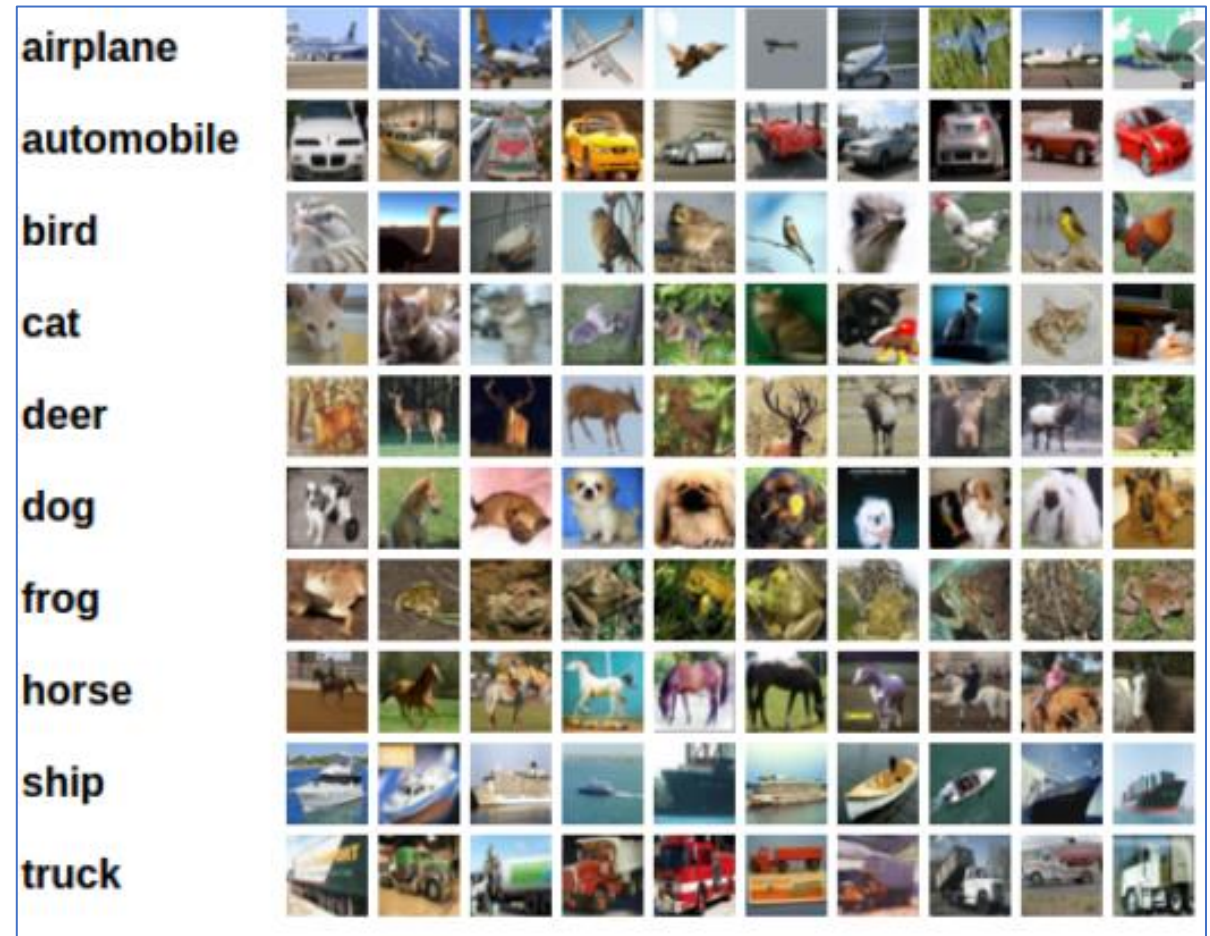
Atelier Pratique AP-ML6 : Exercice ML6.5

Créer un programme de classification d'images : source = dataset **CIFAR10** , nb classes : 10

```
from tensorflow.keras.datasets import cifar10  
  
(x_train, y_train), (x_test, y_test) = cifar10.load_data()
```



SOLUTION



Atelier Pratique AP-ML6 : Quiz

Quiz : Machine Learning, Objectif pédagogique 4

