

Les bibliothèques nécessaires sont importées:

tensorflow : pour utiliser le réseau de neurones
numpy : pour utiliser les tableaux
matplotlib.pyplot : pour afficher les images
random : pour générer des nombres aléatoires

Chargement de l'ensemble de données MNIST (x_train, y_train), (x_test, y_test) :

x_train et x_test contiennent les images.

y_train et y_test contiennent les étiquettes correspondantes de chaque image.

Les images (x_train et x_test) sont redimensionnées pour ne contenir que des pixels noirs ou blancs (0 ou 1) en utilisant un seuil de 127.

La fonction Divide_By_4() divise l'image en quatre parties : top_right, top_left, bottom_right, bottom_left

La fonction Divide_By_4_Again() utilise la fonction Divide_By_4() pour diviser l'image en 16 sous-images.

La fonction reformulate_Mnist_With_Percentage() prend en entrée un ensemble de données (x) et retourne une liste de pourcentage de pixels noirs pour chaque sous-image

reformed_x_train et reformed_x_test utilise la fonction reformulate_Mnist_With_Percentage() pour être reformatées

Les étiquettes (y_train et y_test) sont modifiées pour qu'elles soient binaires : 1 si l'image est un 5, 0 sinon

Le modèle est défini en utilisant keras.Sequential. Il est composé de deux couches :

La première couche est une couche dense avec 6 neurones et une fonction d'activation relu

La deuxième couche est une couche dense avec 1 neurone et une fonction d'activation sigmoid

Le modèle est compilé en utilisant :

l'optimiseur 'adam' pour mettre à jour les poids

la fonction de coût 'BinaryCrossentropy' pour calculer la perte

la métrique 'accuracy' pour évaluer la performance

Le modèle est entraîné sur les données d'entraînement (reformed_x_train) et les étiquettes d'entraînement (y_train_five) pendant 3 epochs.

Enfin, une fonction tester() est définie pour tester les prédictions du modèle sur les données de test.

Il demande à l'utilisateur de saisir un numéro de l'image de test, utilise ce numéro pour accéder à l'image correspondante, et faire la prediction sur cette image.