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.