Edge Intelligence

LAB Assignment-1:

D. Sai Mohith

25MML0032

Topic: Image preprocessing of MNIST dataset

Steps Involved:

1. **Imported required packages and modules:**
   import tensorflow as tf

   from tensorflow.keras.layers import Input,Dense

   from tensorflow.keras.models import Sequential
2. **Loaded MNIST Dataset:**
   (X_train,y_train),(X_test,y_test)=tf.keras.datasets.mnist.load_data()
3. **Dataset Inspection:**

   X_train.shape

   y_train.shape

   X_test.shape

   y_test.shape

   X_train.dtype
4. **Adding channel dimension if images are grayscale:**

   if len(X_train.shape) == 3: # means grayscale images like MNIST

   X_train = X_train[..., None] # add channel dimension → (60000,28,28,1)

   X_test = X_test[..., None]
5. **Verification of standard image size:**

   TARGET_SIZE = (28, 28)

   if X_train.shape[1:3]!=TARGET_SIZE:
   print("Resizing images to 28x28")
   X_train = tf.image.resize(X_train, TARGET_SIZE)
   X_test = tf.image.resize(X_test, TARGET_SIZE)
   else:
   print("Images already 28x28 So no resizing needed.")
6. **Normalized:**

   X_train=X_train/255
7. **Reshaping:**
   X_train=X_train.reshape(-1,784)

Summary:

The assessment starts by importing the fundamental modules needed to manage data and provide basic inputs that are suitable for neural networks. We import the Input and Dense layers, as well as the Sequential model class, from TensorFlow's Keras API. These imports ensure that all necessary tools are available in case they are needed later, even if the code does not yet fully construct a model.

TensorFlow's built-in function is then used to load the MNIST dataset. The handwritten digit pictures in this dataset have previously been separated into training and testing sets.

The dataset is examined before to any processing. The number of samples and the height/width of each picture may be verified by printing shapes like **X_train.shape** and **y_train.shape**. The data type, usually uint8, which indicates that each pixel spans from 0 to 255, may also be seen by checking **X_train.dtype**. Due to its greyscale nature, MNIST pictures were initially just three-dimensional in form (samples, height, and breadth). Even for greyscale photos, an additional channel dimension is preferred by many image-based procedures.

In the event that the array only contains three dimensions, the form is changed to (samples, 28, 28, 1) by adding a new channel axis at the end. As a result, the data is compatible with model layers and image processing that anticipate a channel dimension. The selected goal size is 28 x 28. The method uses TensorFlow's resizing function to enlarge each picture if its existing dimensions do not match this size.

Each picture is reshaped from a 2-D grid into a single continuous row because each 28x28 image has 784 pixels. Simple fully connected neural networks and any algorithm that anticipates one-dimensional input characteristics rather than two-dimensional visuals can benefit from this approach.

All things considered, the pipeline imports the data, verifies its format, adds a channel if necessary, resizes photos as appropriate, normalises pixel values, and reshapes everything into a tidy, consistent structure that is simple to feed into subsequent machine-learning stages.