**PROGRAM [15]:**

# Importing the required libraries

from tensorflow.keras.datasets import mnist

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv2D, MaxPool2D, Flatten, Dense

from tensorflow.keras.layers import Dropout

# Loading data

(X\_train, y\_train), (X\_test, y\_test) = mnist.load\_data()

# Reshaping data

X\_train = X\_train.reshape((X\_train.shape[0], X\_train.shape[1], X\_train.shape[2], 1))

X\_test = X\_test.reshape((X\_test.shape[0], X\_test.shape[1], X\_test.shape[2], 1))

# Checking the shape after reshaping

print(X\_train.shape)

print(X\_test.shape)

# Normalizing the pixel values

X\_train = X\_train / 255

X\_test = X\_test / 255

# Defining model

model = Sequential()

# Adding convolution layer

model.add(Conv2D(32, (3, 3), activation='relu', input\_shape=(28, 28, 1)))

# Adding pooling layer

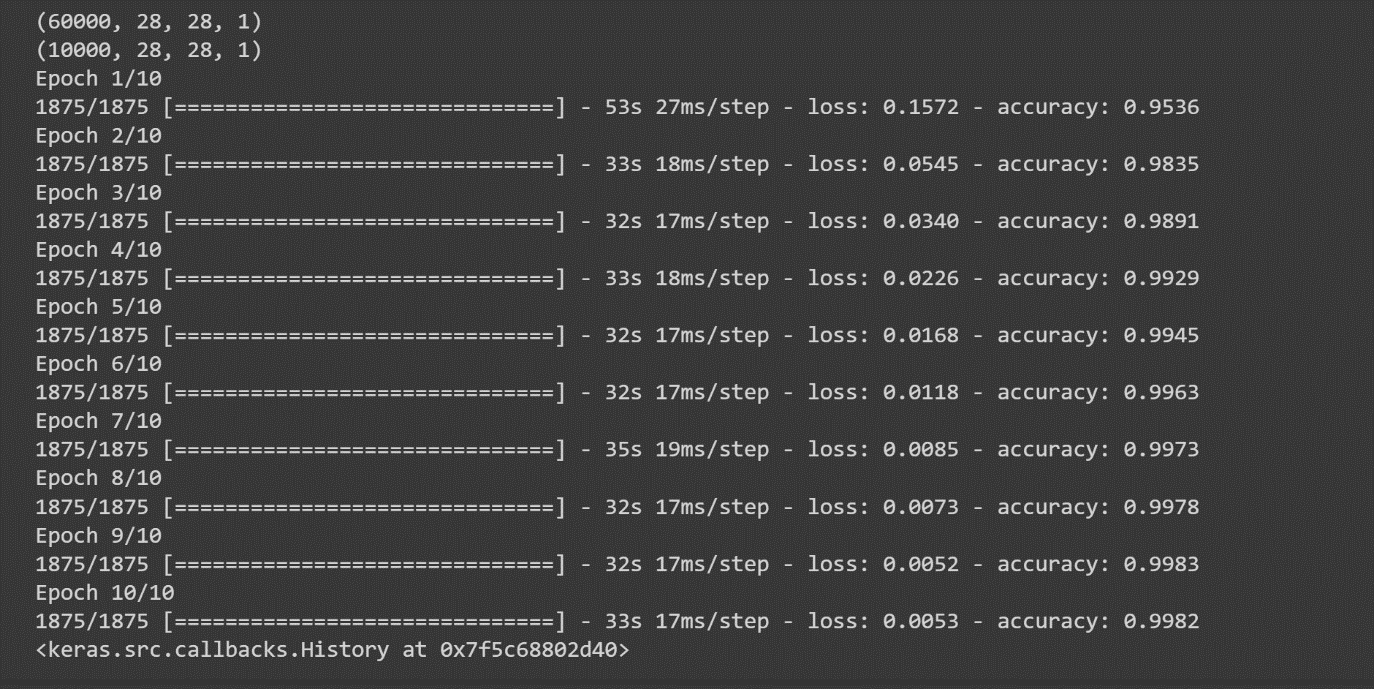
model.add(MaxPool2D(2, 2))

# Adding fully connected layer

model.add(Flatten())

model.add(Dense(100, activation='relu'))

**OUTPUT [15]:**



Adding output layer

model.add(Dense(10, activation='softmax'))

# Compiling the model

model.compile(loss='sparse\_categorical\_crossentropy', optimizer='adam', metrics=['accuracy'])

# Fitting the model

model.fit(X\_train, y\_train, epochs=10)