import keras

from keras.datasets import mnist

from keras.models import Sequential

from keras.layers import Dense,Dropout

batch\_size=8

num\_classes=10

epochs=5

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

print(x\_train.shape)

x\_train = x\_train.reshape(60000, 784)

x\_test = x\_test.reshape(10000, 784)

x\_train = x\_train.astype('float32')

x\_test = x\_test.astype('float32')

print(x\_train.shape)

x\_train = x\_train.reshape(60000, 784)

x\_test = x\_test.reshape(10000, 784)

x\_train = x\_train.astype('float32')

x\_test = x\_test.astype('float32')

print(x\_train.shape)

x\_train /= 255

x\_test /= 255

print(x\_train.shape[0], 'train samples')

print(x\_test.shape[0], 'test samples')

y\_train = keras.utils.to\_categorical(y\_train, num\_classes)

y\_test = keras.utils.to\_categorical(y\_test, num\_classes)

model = Sequential()

model.add(Dense(512, activation='relu', input\_shape=(784,)))

model.add(Dropout(0.2))

model.add(Dense(512, activation='relu'))

model.add(Dropout(0.2))

model.add(Dense(num\_classes, activation='softmax'))

model.summary()

odel.compile(loss='categorical\_crossentropy',

              optimizer='adam',

              metrics=['accuracy'])

history = model.fit(x\_train, y\_train,

                    batch\_size=batch\_size,

                    epochs=epochs,

                    verbose=1,

                    validation\_data=(x\_test, y\_test))