```
import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Flatten, Dropout, Conv2D, MaxPooling2D
from keras import backend as K

# Load the MNIST dataset and split it into training and testing sets
(x_train, y_train), (x_test, y_test) = mnist.load_data()

# Preprocess the data
x_train = x_train.reshape(x_train.shape[0], 28, 28, 1)
x_test = x_test.reshape(x_test.shape[0], 28, 28, 1)
input_shape = (28, 28, 1)

x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train /= 255
x_test /= 255

# Convert labels to one-hot encoding
num_classes = 10
y_train = keras.utils.to_categorical(y_train, num_classes)
y_test = keras.utils.to_categorical(y_test, num_classes)

# Create a Sequential model
model = Sequential()

# Add convolutional layers
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=input_shape))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())

# Add fully connected layers
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))

# Compile the model
model.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=keras.optimizers.Adadelta(),
              metrics=['accuracy'])

# Train the model
model.fit(x_train, y_train,
          batch_size=128,
          epochs=12,
          verbose=1,
          validation_data=(x_test, y_test))
```

```
# Evaluate the model
score = model.evaluate(x_test, y_test, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist
11490434/11490434 [==============================] - 1s 0us/step
Epoch 1/12
469/469 [==============================] - 154s 325ms/step - loss: 2.2741 - accuracy: 0
Epoch 2/12
469/469 [==============================] - 154s 327ms/step - loss: 2.1920 - accuracy: 0
Epoch 3/12
469/469 [==============================] - 149s 318ms/step - loss: 2.0904 - accuracy: 0
Epoch 4/12
469/469 [==============================] - 151s 321ms/step - loss: 1.9549 - accuracy: 0
Epoch 5/12
469/469 [==============================] - 149s 317ms/step - loss: 1.7910 - accuracy: 0
Epoch 6/12
469/469 [==============================] - 154s 329ms/step - loss: 1.6068 - accuracy: 0
Epoch 7/12
469/469 [==============================] - 149s 317ms/step - loss: 1.4273 - accuracy: 0
Epoch 8/12
469/469 [==============================] - 156s 333ms/step - loss: 1.2730 - accuracy: 0
Epoch 9/12
469/469 [==============================] - 155s 331ms/step - loss: 1.1467 - accuracy: 0
Epoch 10/12
469/469 [==============================] - 148s 316ms/step - loss: 1.0454 - accuracy: 0
Epoch 11/12
469/469 [==============================] - 155s 331ms/step - loss: 0.9669 - accuracy: 0
Epoch 12/12
469/469 [==============================] - 155s 331ms/step - loss: 0.9036 - accuracy: 0
Test loss: 0.6499320268630981
Test accuracy: 0.8500000238418579
```