```
from keras.utils import np_utils
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Conv2D, pooling, Flatten, Dense
# MNIST data
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
print(train_images.shape, train_labels.shape, test_images.shape, test_labels.shape)
train_images = train_images.reshape(train_images.shape[0], 28,28,1).astype('float32')/255.0
test_images = test_images.reshape(test_images.shape[0], 28,28,1).astype('float32')/255.0
train_labels = np_utils.to_categorical(train_labels) # One-Hot Encoding
test_labels = np_utils.to_categorical(test_labels) # One-Hot Encoding
# Model
model = Sequential()
model.add(Conv2D(32, (3,3), padding='same', strides=(1,1), activation='relu', input_shape=
(28,28,1)))
print(model.output_shape)
model.add(pooling.MaxPooling2D(pool_size=(2,2)))
print(model.output_shape)
model.add(Conv2D(64, (3,3), padding='same', strides=(1,1), activation='relu'))
print(model.output_shape)
model.add(pooling.MaxPooling2D(pool_size=(2,2)))
print(model.output_shape)
model.add(Flatten())
model.add(Dense(5000, activation='relu'))
model.add(Dense(10, activation='softmax')) # units=10, activation='softmax'
model.compile(loss='categorical_crossentropy', optimizer='sgd', metrics=['accuracy'])
# Training
model.fit(train_images, train_labels, epochs=15, batch_size=100, verbose=1)
# Testing
_, accuracy = model.evaluate(test_images, test_labels)
print('Accuracy: ', accuracy)
model.summary()
```