```
#import mnist
from keras.datasets import mnist


(train_data,train_target),(test_data,test_target)=mnist.load_data()


print(train_data.shape)


print(test_data.shape)


print(train_target[:5])


from matplotlib import pyplot as plt
plt.imshow(train_data[0],cmap='gray')
plt.show()


from keras.models import Sequential
from keras.layers import Dense, Flatten
model= Sequential()
model.add(Flatten(input_shape=(28,28)))
model.add(Dense(512, activation="relu"))
model.add(Dense(256, activation="relu"))
model.add(Dense(128, activation="relu"))
model.add(Dense(64, activation="relu"))
model.add(Dense(10, activation="softmax"))
model.compile(optimizer='adam',loss="categorical_crossentropy",metrics=['accuracy'])
model.summary()


!pip install numpy

from keras.utils import np_utils
new_train_target= np_utils.to_categorical(train_target)
new_test_target = np_utils.to_categorical(test_target)


new_train_data=train_data/255
new_test_data=test_data/255


model.fit(new_train_data,new_train_target,epochs=20)


from matplotlib import pyplot as plt
```

```
plt.plot(model.history.history['accuracy'])
plt.xlabel('epochs')
plt.ylabel('accuracy')
plt.show()
```

```
from matplotlib import pyplot as plt
plt.plot(model.history.history['loss'])
plt.xlabel('epochs')
plt.ylabel('loss')
plt.show()
```

```
model.evaluate(new_test_data,new_test_target)
```