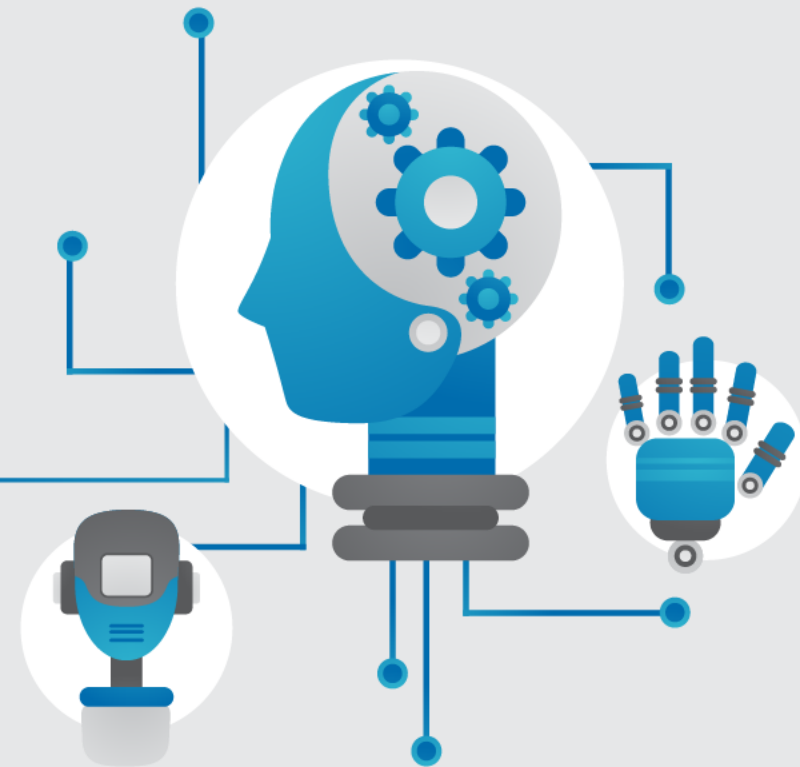


# MNIST手寫字 資料集介紹





# MNIST 資料集



- › MNIST數據庫 ( Modified National Institute of Standards and Technology database ) 是大型的手寫數字數據庫，這個數據庫被廣泛應用於機器學習領域中圖形的訓練和測試。
- › MNIST數據庫是源自於美國國家標準與技術研究所 ( NIST ) 的Special Database 1和Special Database 3。

**Special Database 1**：高中生的手寫字

**Special Database 3**：美國人口普查局員工的手寫字



# MNIST 資料集



- › 下載網址：<http://yann.lecun.com/exdb/mnist/>

Four files are available on this site:

```
train-images-idx3-ubyte.gz: training set images (9912422 bytes)
train-labels-idx1-ubyte.gz: training set labels (28881 bytes)
t10k-images-idx3-ubyte.gz:  test set images (1648877 bytes)
t10k-labels-idx1-ubyte.gz:  test set labels (4542 bytes)
```

- › 訓練資料：60000張；測試資料：10000張
- › 0 ~ 9 手寫數字
- › 每張皆為28x28的灰階影像（像素值0 ~ 255）






# MNIST 資料集下載

機器學習實務




› Kaggle競賽 - Digit Recognizer

<https://www.kaggle.com/c/digit-recognizer/overview>

 test.csv

2019/12/11

 train.csv

2019/12/11

› 從Keras套件下載內建資料集

```
import keras
```

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



# MNIST 圖檔處理



› 圖檔格式：42,000張28X28 jpg圖檔

影像	
影像 ID	
尺寸	28 x 28
寬度	28 個像素
高度	28 個像素
水平解析度	96 dpi
垂直解析度	96 dpi
位元深度	8

0 1 2 3 4 5 6 7 8 9



# MNIST 圖檔處理



## › 資料切割

訓練資料36,000張，測試資料6,000張

## › 圖檔檔名編碼

0.0.jpg, 0.1.jpg, 0.2.jpg, ..., 0.3599.jpg,

1.0.jpg, 1.1.jpg, 1.2.jpg, ..., 1.3599.jpg,

.....

9.0.jpg, 9.1.jpg, 9.2.jpg, ..., 9.3599.jpg



# 讀取圖檔

› 主要步驟

1. 載入函示庫



2. 預留資料空間



3. 讀取**訓練**圖片內容及label



4. 讀取**測試**圖片內容及label



5. 回傳切割結果

機器學習實務







# 讀取圖檔



## › 載入函示庫

```
3 # 載入函示庫os讀取目錄檔名，PIL讀取影像內容，numpy儲存資料  
4 import os  
5 from PIL import Image  
6 import numpy as np
```







# 讀取圖檔



## › 預留資料空間

```
8 # 讀取資料夾mnist下的42000張圖片，圖片為灰階圖(只有1通道)，圖像大小28*28
9 def load_data():
10     # 宣告訓練資料train_data及其標記train_labels，測試資料test_data及其標記test_labels
11     train_data = np.empty((36000,1,28,28),dtype="float32")
12     train_labels = np.empty((36000,),dtype="uint8")
13     test_data = np.empty((6000,1,28,28),dtype="float32")
14     test_labels = np.empty((6000,),dtype="uint8")
```



# 讀取圖檔



## › 讀取訓練圖片內容及label

```
16 # 讀取訓練圖片內容及從檔名切出label
17 imgs_1 = os.listdir("./trainImg")
18 num_1 = len(imgs_1)
19 for i in range(num_1):
20     img_1 = Image.open("./trainImg/"+imgs_1[i])
21     arr_1 = np.asarray(img_1,dtype="float32")
22     train_data[i,:,:,:] = arr_1
23     train_labels[i] = int(imgs_1[i].split('.')[0])
```





# 讀取圖檔



## › 讀取測試圖片內容及label

```
25 # 讀取測試圖片內容及從檔名切出label
26 imgs_2 = os.listdir("./testImg")
27 num_2 = len(imgs_2)
28 for i in range(num_2):
29     img_2 = Image.open("./testImg/"+imgs_2[i])
30     arr_2 = np.asarray(img_2,dtype="float32")
31     test_data[i,:,:,:] = arr_2
32     test_labels[i] = int(imgs_2[i].split('.')[0])
```





# 讀取圖檔



## › 回傳切割結果

```
34 # 回傳結果  
35 return (train_data,train_labels), (test_data,test_labels)
```



# 打亂資料



› 使用函示庫random的shuffle打亂資料

```
16 from data import load_data
17
18 # the data, split between train and test sets
19 (train_data, train_labels), (test_data, test_labels) = load_data()
20
21 import random
22
23 index = [i for i in range(len(train_data))]
24 random.shuffle(index)
25 train_data = train_data[index]
26 train_labels = train_labels[index]
27
28 index = [i for i in range(len(test_data))]
29 random.shuffle(index)
30 test_data = test_data[index]
31 test_labels = test_labels[index]
```

› 使用model.fit，設定shuffle==True進行shuffle

```
model.fit ( train_data, train_labels, epochs=20,
validation_split=0.2, shuffle=True )
```