In [1]:

```
#打开数据集
import numpy as np
import scipy.io as scio
import matplotlib as mpl
import matplotlib.pyplot as plt

file_path ="C:/Users/70951/Desktop/mnist-original.mat"
mnist = scio.loadmat(file_path)
mnist.keys()

#数据整理
X, y = mnist["data"], mnist["label"]
X = X.transpose()
X.shape
y = y.transpose()
y.shape
y = y.astype(np.uint8)
X_train, X_test, y_train, y_test = X[:60000], X[60000:], y[:60000], y[60000:]
y_train=y_train.ravel()
y_test=y_test.ravel()
```

In [2]:

```
#训练模型
from sklearn.svm import SVC
svm_clf =  SVC(C=1.0, kernel='poly', gamma='auto', max_iter=50, random_state=405)
svm_clf.fit(X_train, y_train)
```

```
E:\python\lib\site-packages\sklearn\svm\_base.py:231: ConvergenceWarning: Solver ter
minated early (max_iter=50).  Consider pre-processing your data with StandardScaler
or MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
```

Out[2]:

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto', kernel='poly',
    max_iter=50, probability=False, random_state=405, shrinking=True, tol=0.001,
    verbose=False)
```

In [4]:

```
#k折预测
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import cross_val_predict
cross_val_score(svm_clf, X_train, y_train.ravel(), cv=3, scoring="accuracy")# 每一次验证的正确概率输
```

```
E:\python\lib\site-packages\sklearn\svm\_base.py:231: ConvergenceWarning: Solver ter
minated early (max_iter=50).  Consider pre-processing your data with StandardScaler
or MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
E:\python\lib\site-packages\sklearn\svm\_base.py:231: ConvergenceWarning: Solver ter
minated early (max_iter=50).  Consider pre-processing your data with StandardScaler
or MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
E:\python\lib\site-packages\sklearn\svm\_base.py:231: ConvergenceWarning: Solver ter
minated early (max_iter=50).  Consider pre-processing your data with StandardScaler
or MinMaxScaler.
  % self.max_iter, ConvergenceWarning)
```

Out[4]:

```
array([0.8491 , 0.8412 , 0.85925])
```

In [ ]:

```
#寻优,由于耗时太久，遂放弃
param_grid = [{'C':[1,10,100],
               'kernel':["linear", "poly", "rbf", "sigmoid",],
               'gamma':["auto"],
               'max_iter':[5,30,100]}]
grid_search = GridSearchCV(svm_clf, param_grid, cv=3, verbose=3)
grid_search.fit(X_train, y_train.ravel())
grid_search.best_params_
```