

```
hw1.py
1  from mnist import MNIST
2  import sklearn.metrics as metrics
3  import numpy as np
4  from numpy.linalg import inv
5
6  NUM_CLASSES = 10
7
8  def load_dataset():
9      mndata = MNIST('./data/')
10     X_train, labels_train = map(np.array, mndata.load_training())
11     X_test, labels_test = map(np.array, mndata.load_testing())
12     X_train = X_train/255.0
13     X_test = X_test/255.0
14     return (X_train, labels_train), (X_test, labels_test)
15
16
17 def train(X_train, y_train, reg=0):
18     ''' Build a model from X_train -> y_train '''
19     #here involve a hyper parameter
20     inverse = inv(np.dot(np.matrix.transpose(X_train), X_train) + 0.5*np.identity(784))
21     return np.dot(inverse, np.dot(np.matrix.transpose(X_train), one_hot(y_train)))
22
23 def one_hot(labels_train):
24     '''Convert categorical labels 0,1,2,...,9 to standard basis vectors in  $\mathbb{R}^{10}$ '''
25     return np.array([[1 if i == labels_train[k] else 0 for i in range(10)] for k in range(len(labels_train))])
26
27 def predict(model, X):
28     ''' From model and data points, output prediction vectors '''
29     result = np.dot(np.matrix.transpose(model), np.matrix.transpose(X)) #get a vector
30     return [np.argmax(i) for i in np.matrix.transpose(result)] #single array with dim = 1*60000
31
32 if __name__ == "__main__":
33     (X_train, labels_train), (X_test, labels_test) = load_dataset()
34     model = train(X_train, labels_train)
35     y_train = one_hot(labels_train)
36     y_test = one_hot(labels_test)
37
38     pred_labels_train = predict(model, X_train)
39     pred_labels_test = predict(model, X_test)
40
41     print("Train accuracy: {0}".format(metrics.accuracy_score(labels_train, pred_labels_train)))
42     print("Test accuracy: {0}".format(metrics.accuracy_score(labels_test, pred_labels_test)))
43
```