Nearest Neighbor Classifier

L1 distance:

$$d_1(I_1,I_2)=\sum_p||I_{p1}-I_{p2}||$$

$$
\begin{array}{c}\text{test image}\\ \left|\begin{array}{|c|c|c|c|}\hline 56 & 32 & 10 & 18\\ \hline 90 & 23 & 128 & 133\\ \hline 24 & 26 & 178 & 200\\ \hline 2 & 0 & 255 & 220\\ \hline\end{array}\right.\end{array}
-
\begin{array}{c}\text{training image}\\ \left.\begin{array}{|c|c|c|c|}\hline 10 & 20 & 24 & 17\\ \hline 8 & 10 & 89 & 100\\ \hline 12 & 16 & 178 & 170\\ \hline 4 & 32 & 233 & 112\\ \hline\end{array}\right|\end{array}
=
\begin{array}{c}\text{pixel-wise absolute value differences}\\ \begin{array}{|c|c|c|c|}\hline 46 & 12 & 14 & 1\\ \hline 82 & 13 & 39 & 33\\ \hline 12 & 10 & 0 & 30\\ \hline 2 & 32 & 22 & 108\\ \hline\end{array}\end{array}
\rightarrow 456
$$

```
Xtr, Ytr, Xte, Yte = load_CIFAR10('data/cifar10/') # a magic function we provide
# flatten out all images to be one-dimensional
Xtr_rows = Xtr.reshape(Xtr.shape[0], 32 * 32 * 3) # Xtr_rows becomes 50000 x
3072
Xte_rows = Xte.reshape(Xte.shape[0], 32 * 32 * 3) # Xte_rows becomes 10000 x
3072
```

训练集有50000张图片，每一行是一个图片，一个图片有3072个值

```
nn = NearestNeighbor() # create a Nearest Neighbor classifier class
nn.train(Xtr_rows, Ytr) # train the classifier on the training images and labels
Yte_predict = nn.predict(Xte_rows) # predict labels on the test images
# and now print the classification accuracy, which is the average number
# of examples that are correctly predicted (i.e. label matches)
print 'accuracy: %f' % ( np.mean(Yte_predict == Yte) )
```

```
import numpy as np

class NearestNeighbor(object):
  def __init__(self):
    pass
  def train(self, X, y):
    """ X is N x D where each row is an example. Y is 1-dimension of size N """
    # the nearest neighbor classifier simply remembers all the training data
    self.Xtr = X
    self.ytr = y

  def predict(self, X):
    """ X is N x D where each row is an example we wish to predict label for """
    num_test = X.shape[0]
    # lets make sure that the output type matches the input type
    Ypred = np.zeros(num_test, dtype = self.ytr.dtype)

    # loop over all test rows
    for i in xrange(num_test):          #对于每个测试的图片

      distances = np.sum(np.abs(self.Xtr - X[i,:]), axis = 1)      # 算出与每个训练集图片的距离
      min_index = np.argmin(distances)                             # 选出距离最小的图片的index
      Ypred[i] = self.ytr[min_index]                               # predict the label of the nearest example

    return Ypred
```

L2 distance：

$$d_2(I_1,I_2)=\sqrt{\sum_p \left(I_{p1}-I_{p2}\right)^2}$$

```
distances = np.sqrt(np.sum(np.square(self.Xtr - X[i,:]), axis = 1))
```

Tips for coding

- 直接算`accuracy`
    - `np.mean(Yte_predict == Yte)`

- 广播
    - `distances = np.sum(np.abs(self.Xtr - X[i,:]), axis = 1)`

- 找出最小值的`index`
    - `min_index = np.argmin(distances)`

---

k - Nearest Neighbor Classifier：

k closest images