



In [49]: `%pip install opencv-python`

```
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simpleNote: you may need to re
start the kernel to use updated packages.
Requirement already satisfied: opencv-python in c:\users\xuyic\miniconda3\lib\site-p
ackages (4.6.0.66)
Requirement already satisfied: numpy>=1.14.5 in c:\users\xuyic\miniconda3\lib\site-p
ackages (from opencv-python) (1.21.6)
```

In [50]: `%pip install seaborn`

```
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple
Requirement already satisfied: seaborn in c:\users\xuyic\miniconda3\lib\site-package
s (0.11.2)
Requirement already satisfied: matplotlib>=2.2 in c:\users\xuyic\miniconda3\lib\site
-packages (from seaborn) (3.5.2)
Requirement already satisfied: numpy>=1.15 in c:\users\xuyic\miniconda3\lib\site-pac
kages (from seaborn) (1.21.6)
Requirement already satisfied: pandas>=0.23 in c:\users\xuyic\miniconda3\lib\site-pa
ckages (from seaborn) (1.3.5)
Requirement already satisfied: scipy>=1.0 in c:\users\xuyic\miniconda3\lib\site-pack
ages (from seaborn) (1.7.3)
Requirement already satisfied: cycler>=0.10 in c:\users\xuyic\miniconda3\lib\site-pa
ckages (from matplotlib>=2.2->seaborn) (0.11.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\xuyic\miniconda3\lib\site-p
ackages (from matplotlib>=2.2->seaborn) (9.2.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\xuyic\miniconda3\lib\si
te-packages (from matplotlib>=2.2->seaborn) (3.0.9)
Requirement already satisfied: packaging>=20.0 in c:\users\xuyic\miniconda3\lib\site
-packages (from matplotlib>=2.2->seaborn) (21.3)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\xuyic\miniconda3\lib\si
te-packages (from matplotlib>=2.2->seaborn) (1.4.4)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\xuyic\miniconda3\lib\si
te-packages (from matplotlib>=2.2->seaborn) (4.34.4)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\xuyic\miniconda3\lib
\site-packages (from matplotlib>=2.2->seaborn) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in c:\users\xuyic\miniconda3\lib\site-pa
ckages (from pandas>=0.23->seaborn) (2022.1)
Requirement already satisfied: typing-extensions in c:\users\xuyic\miniconda3\lib\si
te-packages (from kiwisolver>=1.0.1->matplotlib>=2.2->seaborn) (4.3.0)
Requirement already satisfied: six>=1.5 in c:\users\xuyic\miniconda3\lib\site-packag
es (from python-dateutil>=2.7->matplotlib>=2.2->seaborn) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [6]: `%pip install keras`
`%pip install download`



Looking in indexes: <https://pypi.tuna.tsinghua.edu.cn/simple>
 Requirement already satisfied: keras in c:\users\xuyic\miniconda3\lib\site-packages (2.9.0)
 Note: you may need to restart the kernel to use updated packages.
 Looking in indexes: <https://pypi.tuna.tsinghua.edu.cn/simple>
 Note: you may need to restart the kernel to use updated packages.
 Collecting download
 Downloading <https://pypi.tuna.tsinghua.edu.cn/packages/37/45/01e7455a9659528e77a414b222326d4c525796e4f571bbabcb2e0ff3d1f4/download-0.3.5-py3-none-any.whl> (8.8 kB)
 Requirement already satisfied: six in c:\users\xuyic\miniconda3\lib\site-packages (from download) (1.16.0)
 Requirement already satisfied: requests in c:\users\xuyic\miniconda3\lib\site-packages (from download) (2.27.1)
 Requirement already satisfied: tqdm in c:\users\xuyic\miniconda3\lib\site-packages (from download) (4.63.0)
 Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\xuyic\miniconda3\lib\site-packages (from requests->download) (1.26.8)
 Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\xuyic\miniconda3\lib\site-packages (from requests->download) (2.0.4)
 Requirement already satisfied: certifi>=2017.4.17 in c:\users\xuyic\miniconda3\lib\site-packages (from requests->download) (2022.6.15)
 Requirement already satisfied: idna<4,>=2.5 in c:\users\xuyic\miniconda3\lib\site-packages (from requests->download) (3.3)
 Requirement already satisfied: colorama in c:\users\xuyic\miniconda3\lib\site-packages (from tqdm->download) (0.4.4)
 Installing collected packages: download
 Successfully installed download-0.3.5

```
In [46]: # CIFAR10
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neural_network import MLPClassifier
import pandas as pd
import numpy as np
```

```
In [47]: cifar10_zip_path = "./cifar-10-python.tar.gz"
cifar10_path = "./cifar-10-python"
import os
print(os.path.exists(cifar10_zip_path))
print(os.path.exists(cifar10_path))
```

True
True

```
In [55]: # load the data
def unpickle(file):
    import pickle
    with open(file, 'rb') as fo:
        dict = pickle.load(fo, encoding='bytes')
    return dict

def load_cifar10_data(cifar10_path):
    xs = []
    ys = []
    for i in range(1, 6):
        data_dict = unpickle(cifar10_path + "/data_batch_" + str(i))
        xs.append(data_dict[b'data'])
        ys.append(data_dict[b'labels'])
    x_train = np.concatenate(xs)
    y_train = np.concatenate(ys)
    del xs, ys
    data_dict = unpickle(cifar10_path + "/test_batch")
    x_test = data_dict[b'data']
```

```

y_test = np.array(data_dict[b'labels'])
return x_train, y_train, x_test, y_test

```

```
In [68]: x_train, y_train, x_test, y_test = load_cifar10_data(cifar10_path)
```

```
In [69]: print(len(x_train), len(y_train), len(x_test), len(y_test))

50000 50000 10000 10000
```

```
In [70]: print(x_train[0].shape)
print(x_train[0])

(3072,)
[ 59  43  50 ... 140  84  72]
```

```
In [71]: # 我们将会使用层级式k均值对这些图像块建立一中随机地提取这些块
# 并且应该为每个训练图像提取两个块
from random import random

extracted_data_set = []
for i in x_train:
    # random between 0 to 2047
    random_index = int(random() * 2069)
    temp = i[random_index:random_index + 2]
    # print(temp)
    extracted_data_set.append(temp)
```

```
In [53]: print(extracted_data_set[:20])
len(extracted_data_set)

[array([117, 120], dtype=uint8), array([61, 93], dtype=uint8), array([29, 36], dtype=
uint8), array([75, 75], dtype=uint8), array([167, 165], dtype=uint8), array([82, 9
5], dtype=uint8), array([54, 44], dtype=uint8), array([146, 176], dtype=uint8), arra
y([178, 177], dtype=uint8), array([63, 77], dtype=uint8), array([47, 54], dtype=uint
8), array([23, 9], dtype=uint8), array([39, 66], dtype=uint8), array([7, 8], dtype=
uint8), array([175, 172], dtype=uint8), array([188, 222], dtype=uint8), array([71, 7
5], dtype=uint8), array([108, 112], dtype=uint8), array([229, 199], dtype=uint8), ar
ray([183, 182], dtype=uint8)]
50000
```

```
Out[53]:
```

```
In [59]: # KNN分类
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

# 将该数据集聚类成50个中心
model = KNeighborsClassifier(n_neighbors=50)
model.fit(extracted_data_set, y_train)
```

```
Out[59]: KNeighborsClassifier(n_neighbors=50)
```

```
In [64]: extracted_test_set = []
for i in x_test:
    # random between 0 to 2047
    random_index = int(random() * 2047)
    temp = i[random_index:random_index + 2]
    # print(temp)
    extracted_test_set.append(temp)

extracted_test_set[:20]
model.score(extracted_test_set, y_test)
```

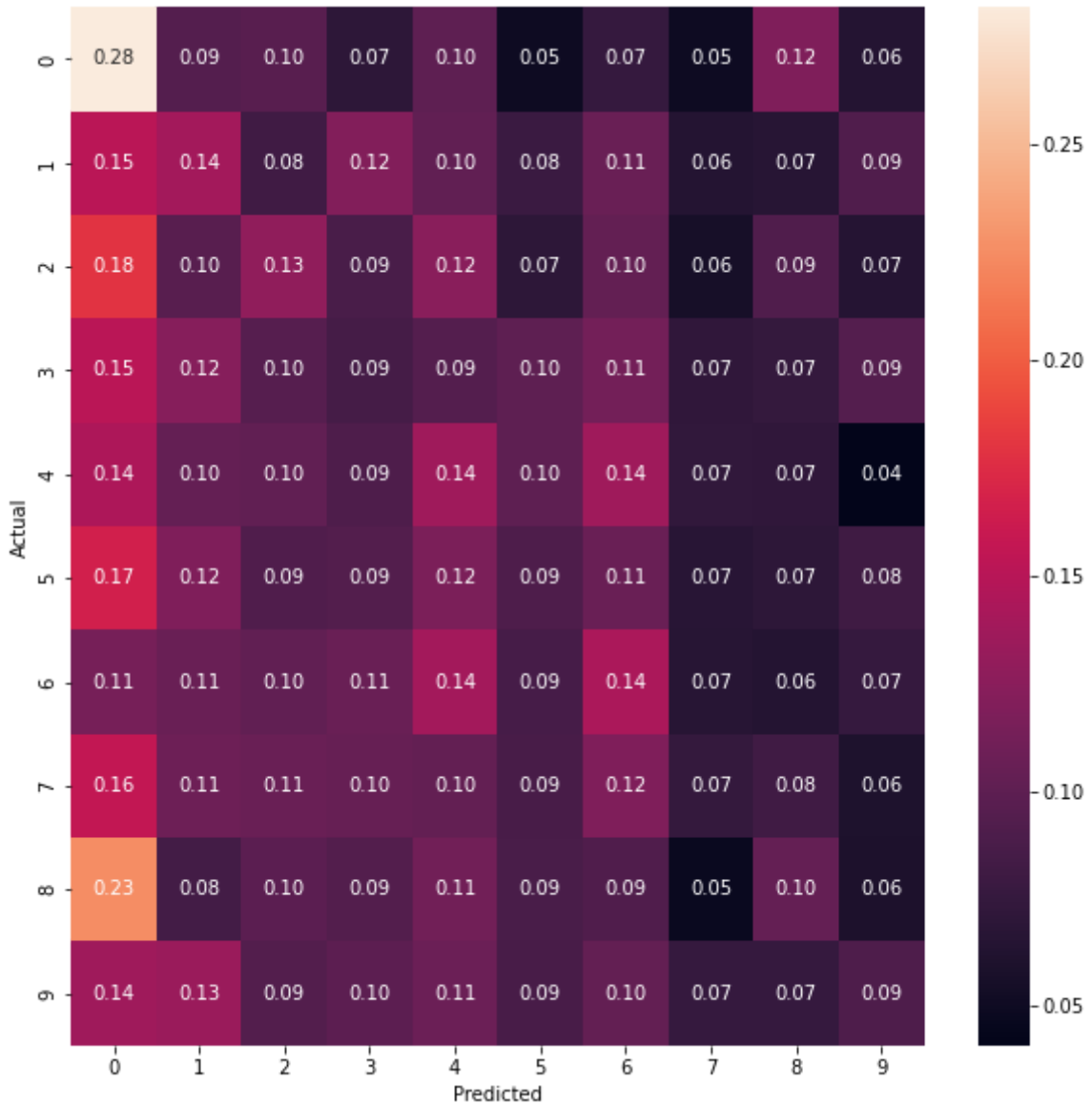
```

score = accuracy_score(y_test, model.predict(extracted_test_set))
print(score)
# plot
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test, model.predict(extracted_test_set))
cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
plt.figure(figsize=(10, 10))
sns.heatmap(cm, annot=True, fmt='.2f', xticklabels=range(10), yticklabels=range(10))
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.show()

```

0.1271



```

In [80]: # 你现在拥有一个含有2500个元素的字典
new_test_set = x_test[:2500]
new_test_set2 = []
new_test_set_label = y_test[:2500]
print(len(new_test_set), len(new_test_set_label))
print(new_test_set[:20], new_test_set_label[:20])

for i in range(2500):
    random_index = int(random() * 2047)
    temp = new_test_set[i][random_index:random_index + 2]
    new_test_set2.append(temp)

```

```
print(new_test_set2[:20], new_test_set_label[:20])

2500 2500
[[158 159 165 ... 124 129 110]
 [235 231 232 ... 178 191 199]
 [158 158 139 ... 8 3 7]
 ...
 [ 60 69 71 ... 198 195 196]
 [223 223 225 ... 79 84 118]
 [ 55 51 50 ... 154 160 148]] [3 8 8 0 6 6 1 6 3 1 0 9 5 7 9 8 5 7 8 6]
[array([166, 132], dtype=uint8), array([178, 165], dtype=uint8), array([233, 238], d
type=uint8), array([201, 202], dtype=uint8), array([113, 107], dtype=uint8), array
([148, 156], dtype=uint8), array([ 59, 114], dtype=uint8), array([106, 104], dtype=u
int8), array([130, 114], dtype=uint8), array([34, 75], dtype=uint8), array([132, 13
3], dtype=uint8), array([158, 230], dtype=uint8), array([16, 15], dtype=uint8), arra
y([247, 247], dtype=uint8), array([23, 34], dtype=uint8), array([113, 113], dtype=ui
nt8), array([242, 244], dtype=uint8), array([171, 136], dtype=uint8), array([ 69, 11
9], dtype=uint8), array([36, 29], dtype=uint8)] [3 8 8 0 6 6 1 6 3 1 0 9 5 7 9 8 5 7
8 6]
```

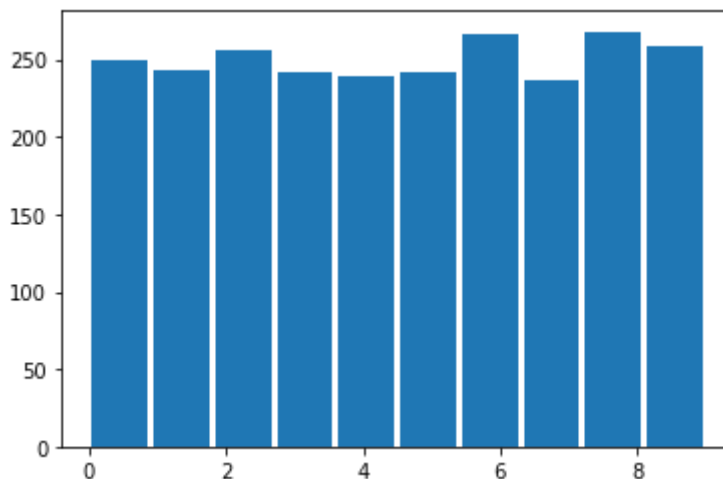
```
In [83]: # predict
print(model.predict(new_test_set2))
print(model.score(new_test_set2, new_test_set_label))

# plot
import matplotlib.pyplot as plt
import seaborn as sns
# 直方图

plt.hist(new_test_set_label, bins=10, rwidth=0.9)
```

```
[8 2 0 ... 2 6 1]
0.1304
```

```
Out[83]: (array([250., 243., 256., 242., 239., 242., 266., 236., 268., 258.]),
 array([0. , 0.9, 1.8, 2.7, 3.6, 4.5, 5.4, 6.3, 7.2, 8.1, 9. ]),
 <BarContainer object of 10 artists>)
```



```
In [ ]:
```