

知能システム レポート 2

三浦夢生

2020 年 11 月 12 日

1 目的

MNIST データセットを用いて数字認識の学習を行う。

2 用語

2.1 MNIST

MNIST とは手書き数字画像のデータセットのことであり、機械学習の分野では非常に広く知られている。画像認識の学習やベンチマークテストに用いられる人気のデータセットである。

2.2 K 近傍法

K 近傍法 (K-Nearest Neighbor method) とは、トレーニングデータを二次元空間上にプロットし、未知のデータをプロットした際、近くの K 個のデータのうち最も多いラベルがつけられる分類の手法である。このとき、近くの K 個を選ぶのには一般的にユークリッド距離 (いわゆる距離) が用いられる。特に、 $k = 1$ のとき最近傍法と呼ばれる。

3 方法

Google Colaboratory 上で MNIST データセットを用いて機械学習を行う。データは <https://www.openml.org/d/554> のデータを用いた。

4 実行結果

今回作成したソースコードを付録に示す。また、実行結果を以下に示す。学習したモデルに付録に示す画像を分類させてみると、正しい結果を出すことが出来た。

```
1 SVC(C=10, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
2     decision_function_shape='ovr', degree=3, gamma=0.001, kernel='rbf',
3     max_iter=-1, probability=False, random_state=None, shrinking=True,
4     tol=0.001, verbose=False)
5 score 0.9888888888888889
6 predict = [1]
```

```
1 KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
2                      metric_params=None, n_jobs=None, n_neighbors=1, p=2,
3                      weights='uniform')
4 score 0.9805555555555555
5 predict = [1]
```

5 付録

今回作成したソースコードを以下に示す.

5.1 SVC+KFoldCV+GS

```
1 from sklearn.model_selection import train_test_split
2 from sklearn.model_selection import KFold
3 from sklearn.model_selection import GridSearchCV
4 from sklearn import datasets, svm, metrics
5 import warnings
6 import cv2
7 import matplotlib.pyplot as plt
8
9 warnings.simplefilter("ignore")
10
11 digits = datasets.load_digits()
12 x = digits.images
13 y = digits.target
14 x = x.reshape((-1, 64))
15 x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2)
16
17 parameters = [
18     {"C": [1, 10, 100, 1000], "kernel": ["linear"]},
19     {"C": [1, 10, 100, 1000], "kernel": ["rbf"], "gamma": [0.001,
20         0.0001]},
21     {"C": [1, 10, 100, 1000], "kernel": ["sigmoid"], "gamma": [0.001,
22         0.0001]}
23 ]
24
25 kfold_cv = KFold(n_splits=5, shuffle=True)
26 clf = GridSearchCV(svm.SVC(), parameters, cv=kfold_cv)
27 clf.fit(x_train, y_train)
28
29 y_pred = clf.predict(x_test)
30 print(clf.best_estimator_)
31 print('score', accuracy_score(y_test, y_pred))
32
33 file_name = "number_1.png"
34
35 img = cv2.imread(file_name)
36 img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
37 img = cv2.resize(img, (8, 8))
38 img = img.reshape((-1, 64))
```

```
37 res = clf.predict(img)
38 print('predict =', res)
```

5.2 KNN+KFoldCV+GS

```
1 from sklearn.model_selection import train_test_split
2 from sklearn.model_selection import KFold
3 from sklearn.model_selection import GridSearchCV
4 from sklearn.neighbors import KNeighborsClassifier
5 from sklearn import datasets, metrics
6 import warnings
7 import cv2
8 import matplotlib.pyplot as plt
9
10 warnings.simplefilter("ignore")
11
12 digits = datasets.load_digits()
13 x = digits.images
14 y = digits.target
15 x = x.reshape((-1, 64))
16 x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2)
17
18 parameters = [
19     {"n_neighbors": [1, 5, 10], "algorithm": ["auto"]},
20     {"n_neighbors": [1, 5, 10], "algorithm": ["ball_tree"], "leaf_size":
21         [10, 20, 30]},
22     {"n_neighbors": [1, 5, 10], "algorithm": ["kd_tree"], "leaf_size": [10,
23         20, 30]}
24 ]
25 kfold_cv = KFold(n_splits=5, shuffle=True)
26 clf = GridSearchCV(KNeighborsClassifier(), parameters, cv=kfold_cv)
27 clf.fit(x_train, y_train)
28
29 y_pred = clf.predict(x_test)
30 print(clf.best_estimator_)
31 print('score', accuracy_score(y_test, y_pred))
32
33 file_name = "number_1.png"
34
35 img = cv2.imread(file_name)
36 img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
37 img = cv2.resize(img, (8, 8))
38 img = img.reshape((-1, 64))
39 res = clf.predict(img)
40 print('predict =', res)
```

5.3 データセットの一部を表示

```
1 import matplotlib.pyplot as plt
2 from sklearn import datasets
3
4 digits = datasets.load_digits()
5
6 for i in range(15):
7     plt.subplot(3, 5, i+1)
8     plt.axis("off")
9     plt.title(str(digits.target[i]))
10    plt.imshow(digits.images[i], cmap="gray")
11
12 plt.show()
```

5.4 分類させた画像

