

인공지능 과제 리포트

과제 제목: KNN with MNIST Data

학번: B511226

이름: 홍현승

## 1. 과제 개요

70000개의 데이터로 이루어진 Mnist data -> 60000개의 Training data( $x_{train}(784 * 60000 \text{ 행렬})$ )과, 각각의 training data의 label(0~9)에 해당하는  $t_{train}(1 * 60000 \text{ 행렬})$ , 10000개의 Test data( $x_{test}(784 * 10000 \text{ 행렬})$ )와 각각의 test data의 label(0~9)에 해당하는  $t_{test}(1 * 10000 \text{ 행렬})$ 로 분할. 1개의 mnist data는  $784(28 * 28)$ 개의 feature을 갖고 있고 각각의 label은 0~9사이의 숫자 중 하나이다.

Training data 60000개를 공간상에 찍어두고, 10000개의 test 데이터중 100개의 데이터를 뽑아 각각의 데이터가 60000개의 데이터중 어느 데이터와 가장 근접한지 weighted KNN으로 판단하여 출력해보는 과제

## 2. 구현 환경

OS: Window

Integrated Development Environment: Jupyter

Using library: math / mnist / numpy/ PIL

## 3. 알고리즘에 대한 설명

KNN 알고리즘은 기존 데이터(training data)를 학습 시킨 후, 이 데이터중 test data가 가장 가까운 부류가 무엇인지 선택하는 알고리즘이다. 이때 distance 를 기준으로 가까운 순서대로 K개(3,5,7)를 선정, 거리에 따른 가중치를 적용한 투표로 Label을 분류한다.

이번 Training data는 feature 개수가 784인 배열이므로, KNN 알고리즘을 그대로 사용하는 경우 Inference에 할애되는 시간이 방대하다.

이를 줄이기 위해 Hand\_Craft라는 기법을 사용한다.

Hand\_Craft는 인간이 직접 개입해 input feature를 가공하는 것을 의미한다.

숫자 이미지(784개의 feature)를 우선  $24*24$  형태의 2차원 배열로 만든 후, 각각의 행(24개)에서 0이 아닌 값의 개수가 몇 개 인지 계산하여  $24*1$ 형태의 행렬로 변환 할 것이다.

(그림 삽입)

## 4. 데이터에 대한 설명

### 4.1 Input Feature

70000 Image Data : 60000 Training data / 10000 Test data

### 4.2 Target Output

n개의(필자는 100개로 지정) testdata label가 n개의 weighted knn의 예측 label을 비교하여 정확도를 계산한다. 이때 예측 label은 두가지 종류로 나뉜다. 첫 번째로, 784개의 feature들을 모두 포함하여 계산한 예측 label, 두 번째로, 784개의 feature를 축약하여 24개의 feature로 만든 후, 계산 한 예측 label이다.

## 1.Hand\_craft과정(reshape) 설명

[illegible]

reshape  $\frac{1}{2} \times 2$   $\frac{1}{2} \times 2$  | handcraft을 하겠다.

```
new_xtrain = []
for i in range(60000):
    new_xtrain.append(reshape(x_train[i]))
```

$x\_train \in 784 \times 60000$  인데  
 $60000 > 4$  이므로 reshape 해서  
 $new\_x\_train$  안에 넣어 주어야 한다.

ex)  $284 \times 60000$   
handout  $\downarrow$   
 $28 \times 60000$

```
new_xtest = []
for i in range(10000):
    new_xtest.append(reshape(x_test[i]))
```

new\_x\_test.append(reshape(x\_test[1]))

⌋

x\_test은 784 x 1000인데,  
각 28x28 픽셀이니까!!

[illegible]

⇒ 28x28 로 reshape  
⇒ 1행에서, 28개의 숫자 "0"이 카운트!! (원래)

```
sh = reshape(x_train[0])
print(sh)
```

0 0 0 0 12 16 16 11 9 5 4 4 6 6 6 5 4 7 8 9 10 10

$$\Rightarrow 0 \mid \frac{2}{3} \mid 1 \mid \text{바탕!!}$$

↓  
1행에서 0이 아닌 값  
↓  
2행에서 0이 아닌 값

## 2.weighted majority vote에 대한 설명

ex) new\_xtrain, new\_xtest[0] (60000, )로, new\_xtrain의 label을

```
def weighted_classify(train_data, test_data, k, train_label):
    dist = []
    vote = [0,0,0,0,0,0,0,0,0]
    bias = 0.1
    for i in range(len(train_data)):
        dist.append(distance(train_data[i], test_data))
    dist = np.array(dist)
    temp = dist.argsort()
    if k == 3:
        for i in range(k):
            vote[train_label[temp[i]]] += (1/dist[temp[i]]+bias) #주석 1번
        result = vote.index(max(vote))
        return result
    elif k == 5:
        for i in range(k):
            vote[train_label[temp[i]]] += (1/dist[temp[i]]+bias)
        result = vote.index(max(vote))
        return result
    else: # k 가 5보다 큰 경우 일때
        for i in range(k):
            vote[train_label[temp[i]]] += (1/dist[temp[i]]+bias)
        result = vote.index(max(vote))
        return result
```

거리가 작은 것부터 정렬된다.

ex) new\_xtest[0]과 new\_xtrain[0] 사이의 거리를 구한 후 append  
new\_xtest[0]과 new\_xtrain[1] 사이의 거리를 구한 후 append  
...

ex) 가장 작은 거리 3개가 0.1, 0.2, 0.3 이리 가장하면 2번씩 더하기  
5번씩 더하기

Vote[2]에  $\frac{1}{0.1} + 0.1$   
+  
 $\frac{1}{0.2} + 0.1$

Vote[5]에  $\frac{1}{0.3} + 0.1$

은해준다.

=> Vote  $\checkmark$   
[0, 0,  $0.2 + \frac{1}{0.1} + 0.1$ , 0, 0,  $\frac{1}{0.3} + 0.1$ , 0.0... ] 이리

Vote 4에 큰 값이 index (2)  
result에 2-랑!

## 6. 결과 및 분석

### 결과 출력 및 결과에 대한 분석

1) No Hand Craft

K = 3인 경우

예측 정확도 : 0.97

시간:2분 32초

```
Test Data Index: 4248 computed class: 2 True class: 2
Test Data Index: 7944 computed class: 1 True class: 1
Test Data Index: 202 computed class: 1 True class: 1
Test Data Index: 7516 computed class: 0 True class: 0
Test Data Index: 1194 computed class: 7 True class: 7
Test Data Index: 8871 computed class: 7 True class: 7
Test Data Index: 5294 computed class: 4 True class: 4
Test Data Index: 1232 computed class: 4 True class: 9
Test Data Index: 1883 computed class: 7 True class: 7
Test Data Index: 7818 computed class: 4 True class: 4
Test Data Index: 3253 computed class: 1 True class: 1
Test Data Index: 1576 computed class: 7 True class: 7
Test Data Index: 6138 computed class: 4 True class: 4
Test Data Index: 8812 computed class: 4 True class: 4
Test Data Index: 8010 computed class: 3 True class: 3
Test Data Index: 8709 computed class: 4 True class: 4
Test Data Index: 1601 computed class: 3 True class: 3
accuracy: 0.97
```

K = 5인 경우

예측 정확도 : 0.98

시간:2분 22초

```
Test Data Index: 7516 computed class: 0 True class: 0
Test Data Index: 1194 computed class: 7 True class: 7
Test Data Index: 8871 computed class: 7 True class: 7
Test Data Index: 5294 computed class: 4 True class: 4
Test Data Index: 1232 computed class: 9 True class: 9
Test Data Index: 1883 computed class: 7 True class: 7
Test Data Index: 7818 computed class: 4 True class: 4
Test Data Index: 3253 computed class: 1 True class: 1
Test Data Index: 1576 computed class: 7 True class: 7
Test Data Index: 6138 computed class: 4 True class: 4
Test Data Index: 8812 computed class: 4 True class: 4
Test Data Index: 8010 computed class: 3 True class: 3
Test Data Index: 8709 computed class: 4 True class: 4
Test Data Index: 1601 computed class: 3 True class: 3
accuracy: 0.98
```

K = 7인 경우

예측 정확도 : 0.98

시간:2분 22초

```
Test Data Index: 1194 computed class: 7 True class: 7
Test Data Index: 8871 computed class: 7 True class: 7
Test Data Index: 5294 computed class: 4 True class: 4
Test Data Index: 1232 computed class: 9 True class: 9
Test Data Index: 1883 computed class: 7 True class: 7
Test Data Index: 7818 computed class: 4 True class: 4
Test Data Index: 3253 computed class: 1 True class: 1
Test Data Index: 1576 computed class: 7 True class: 7
Test Data Index: 6138 computed class: 4 True class: 4
Test Data Index: 8812 computed class: 4 True class: 4
Test Data Index: 8010 computed class: 3 True class: 3
Test Data Index: 8709 computed class: 4 True class: 4
Test Data Index: 1601 computed class: 3 True class: 3
accuracy: 0.98
```

2) Hand Crafted

K = 3인 경우

예측 정확도 : 0.81

시간:1분 45초

```
Test Data Index: 90 computed class: 4 True class: 4
Test Data Index: 1365 computed class: 2 True class: 2
Test Data Index: 3839 computed class: 3 True class: 2
Test Data Index: 7750 computed class: 3 True class: 3
Test Data Index: 5690 computed class: 6 True class: 6
Test Data Index: 2375 computed class: 4 True class: 4
Test Data Index: 8159 computed class: 1 True class: 1
Test Data Index: 1762 computed class: 8 True class: 0
Test Data Index: 6182 computed class: 1 True class: 1
Test Data Index: 496 computed class: 9 True class: 9
accuracy: 0.81
```



K = 5인 경우

예측 정확도 : 0.82

시간:1분 38초

```
Test Data Index: 95 computed class: 4 True class: 4
Test Data Index: 1365 computed class: 2 True class: 2
Test Data Index: 3839 computed class: 2 True class: 2
Test Data Index: 7750 computed class: 3 True class: 3
Test Data Index: 5690 computed class: 6 True class: 6
Test Data Index: 2375 computed class: 4 True class: 4
Test Data Index: 8159 computed class: 1 True class: 1
Test Data Index: 1762 computed class: 8 True class: 0
Test Data Index: 6182 computed class: 1 True class: 1
Test Data Index: 496 computed class: 9 True class: 9
accuracy: 0.8200000000000001
```

K = 7인 경우

예측 정확도 : 0.82

시간:1분 59초

```
Test Data Index: 95 computed class: 4 True class: 4
Test Data Index: 1365 computed class: 2 True class: 2
Test Data Index: 3839 computed class: 2 True class: 2
Test Data Index: 7750 computed class: 3 True class: 3
Test Data Index: 5690 computed class: 6 True class: 6
Test Data Index: 2375 computed class: 4 True class: 4
Test Data Index: 8159 computed class: 1 True class: 1
Test Data Index: 1762 computed class: 0 True class: 0
Test Data Index: 6182 computed class: 1 True class: 1
Test Data Index: 496 computed class: 9 True class: 9
accuracy: 0.8200000000000001
```

분석:

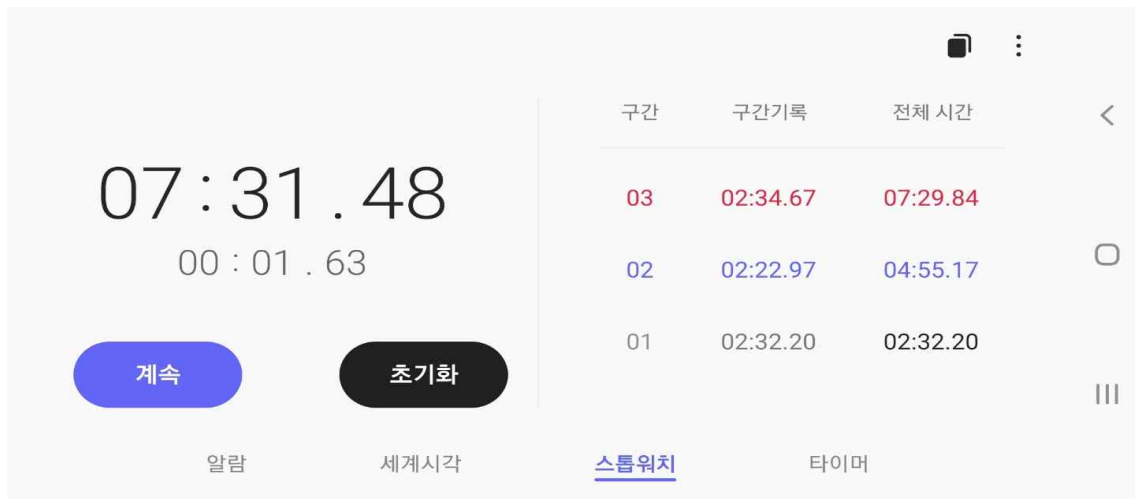
05:23.97

00:00.57

계속초기화

알람세계시각스톱워치타이머

구간	구간기록	전체 시간
03	01:59.05	05:23.40
02	01:38.43	03:24.35
01	01:45.92	01:45.92



위 사진은 handcrafted일 때 아래서부터  $k = 3, 5, 7$ 일 때 걸린시간

아래 사진은 non handcrafted 일 때 아래서부터  $k = 3, 5, 7$ 일 때 걸린시간이다.

시간은 약 1분정도 단축된 것을 알 수 있고, 정확도는 약 0.15정도 내려간 것을 볼 수 있다.