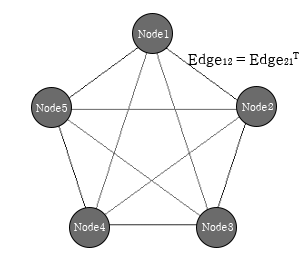
**Project 1: Implementation of k-NN Classifier**

**<Medical Artificial Intelligence, Spring, 2018>**

BME 2013250622 이준재

**1. Hyperparameter tuning by Cross-validation**

 Hyperparameter tuning 을 위한 5-Fold cross-validation 모델을 그래프로 구조화하였다. 전체 Train data 는 같은 크기를 가진 5개의 dataset 으로 랜덤하게 분배되고(로드된 5개의 데이터셋이 랜덤하게 분배되어 있다고 가정) 각 dataset 을 node 로 하는 directed graph 를 만들 수 있다. 이때 연결되는 edge 는 20개인데 Edge(ij) 는 node(i) 를 train data 로 하고 node(j) 를 test data 로 하는 것으로 정의한다. 따라서 Edge(ij) = transpose(Edge(ji)) 가 성립하게 되므로 결과적으로 10개의 Edge connection 의 조합으로 5-fold cross validation 을 수행할 수 있다. 이러한 그래프 모델을 가져온 이유는 알고리즘의 시간 복잡도를 최소화하기 위함이고, 10개의 Edge 조합에 의한 데이터셋은 가능한 최소한의 계산을 요구한다. 예를 들어 5개 중 node1을 test data 로 사용하는 fold dataset 의 컨셉은 다음과 같을 것이다. Fold1 = array(Edge21, Edg31, Edg41, Edg51) = array(Edge12.T, Edge13.T, Edge14.T, Edge15.T) (T는 전치행렬)

다음으로, Edges 의 조합으로 생성된 dataset 의 구조를 시각화하는 과정이다.

|  |  |
| --- | --- |
| 1 | Image(1 by n) |
| 2 | Image(1 by n) |
| .  .  . | .  .  . |
| m | Image(1 by n) |

|  |  |
| --- | --- |
| 1 | Image’(1 by n) |
| 2 | Image’(1 by n) |
| .  .  . | .  .  . |
| m | Image’(1 by n) |

Node1 (train) Node2 (test)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| m by m | Image’ | Image’ | ∙∙∙ | Image’ |
| Image | Distance metric | ∙∙∙ |  |  |
| Image | ∙  ∙  ∙ |  |  |  |
| ∙  ∙  ∙ |  |  |  |  |
| Image |  |  |  |  |

Edge12

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4m by m | Image | Image | ∙∙∙ | Image |
| Image | Distance metric | ∙∙∙ |  |  |
| Image | ∙  ∙  ∙ |  |  |  |
| ∙  ∙  ∙ |  |  |  |  |
| Image |  |  |  |  |

val1 = array(Edge21, Edg31, Edg41, Edg51) < node1 을 test 로 하는 fold

이렇게 distance metric 을 모든 어레이에 대해 수행한 것이 edge 또는 val 이다. 이제 val 들을 sorting 하여 각 열마다 가장 가까운 distance 를 가지는 행의 distance value 와 index number 을 취한다. 이를 위한 코드를 project1\_main cval\_5f function 에서 확인할 수 있다.

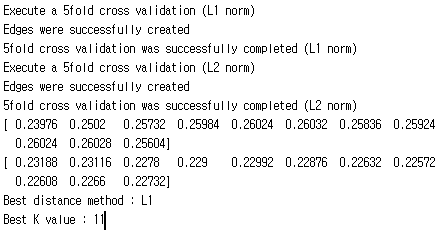
다음으로 sorting 된 행렬에서 k 개의 데이터를 가져와 voting 하는 method 이다. Voting 을 하는 방법은 여러가지가 있을 수 있는데 이 모델에서는 과반수 분류에 의해 발생하는 문제점 중 하나인 ‘outlayer 에 지나치게 편중될 수 있다’ 라는 점을 해결하기 위해 각 vote 에 자신의 distance 에 반비례하는 가중치를 주는 method 를 구현하였다. 이를 위한 코드가 cval\_5f function 에 내장되어 있다.(여기서 0.001 을 더한 후 나눈 이유는 distance 가 zero 로 두 이미지가 완전히 일치하는 경우를 고려하기 위해서이다. CIFAR-10 가 정수형 데이터셋이기 때문에 가장 작은 distance 라도 1이상이고 0.001 은 이에 비해 매우 작다. 또한 이 작업의 장점은 이미지가 완전히 일치하는 경우 vote 권한이 기하급수적으로 상승하기 때문에 같은 이미지를 같다고 판별할 확률이 비약적으로 상승한다.) 또한 가중치 voting 을 위해 val 변수를 매 요소마다(test의 갯수만큼) 인덱스 라벨링시켜 만들어지는 val\_index 변수는 다음과 같다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0 | 1 | ∙∙∙ | 9 |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| ∙  ∙  ∙ |  |  |  |  |
| m |  |  |  |  |

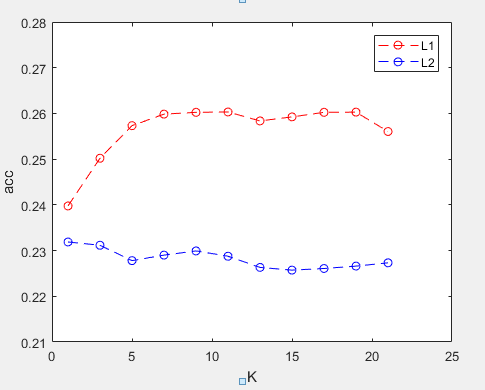
val\_index 이다. 각 행마다 k 번의 voting 이 이루어지고 가중치를 포함해 라벨링된다.

이제 val\_index 에서 각 행마다 가장 큰 값을 취하면 predict labels 을 얻을 수 있고 해당 검증셋 및 k value 에 의한 accuracy 를 얻을 수 있다. 물론 5개의 검증셋 모두 동시에 이 작업이 수행되므로 최종적으로는 각 k 에 대해 5개의 검증셋에서 accuracy 를 평균한 것이 cval\_5f function 의 최종 output 이 된다.

최적의 hyperparameter, 즉 K value 와 distance metric 을 찾기 위한 코드에 cval\_5f\_L1 function 과 cval\_5f\_L2 function 이 포함되고 return 된 accuracy 를 비교하여 k\_best 및 L\_best 값을 print 한다. 결과는 다음과 같다.



위의 결과에서 다양한 k 값 [1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21] 과 L1(위), L2(아래) distance 에 의한 검증 accuracy 를 확인할 수 있다. 이를 플로팅하면 아래와 같다.



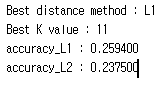
L1 에서 K가 증가할수록 acc 가 커지다가 어느순간 saturation 되고 다시 감소하는 것을 확인할 수 있는데 데이터 개수에 비해 k 값이 너무 커지게 되면 오히려 acc 가 감소하게 된다.

2. k – Nearest Neighbor result

K value : 11

Distance metric : L1

Result



Accuracy : 0.259400