Image Classification

: 컴퓨터가 입력 이미지를 보고 정해놓은 category 집합에서 어떤 category에 속할지 고르는 것.

- 컴퓨터에게 이미지란 단지 거대한 격자 모양의 숫자 집합에 불과하므로 어떤 category인지 구분하기 어렵다. (semantic gap)
- 카메라 위치 변화, 조명의 변화, 객체 자체에 변화를 줬을 때 classification 어려움.

❖ Data-Driven Approcach (데이터 중심 접근 방법)
: 방대한 데이터를 이용해 Machine Learning Classifier를 학습 => 어떤 식으로든 데이터를 잘 요약해서는 다양한 객체들을 인식할 수 있는 모델 만들어냄

Data-Driven Approach

1. Collect a dataset of images and labels
2. Use Machine Learning to train a classifier
3. Evaluate the classifier on new images

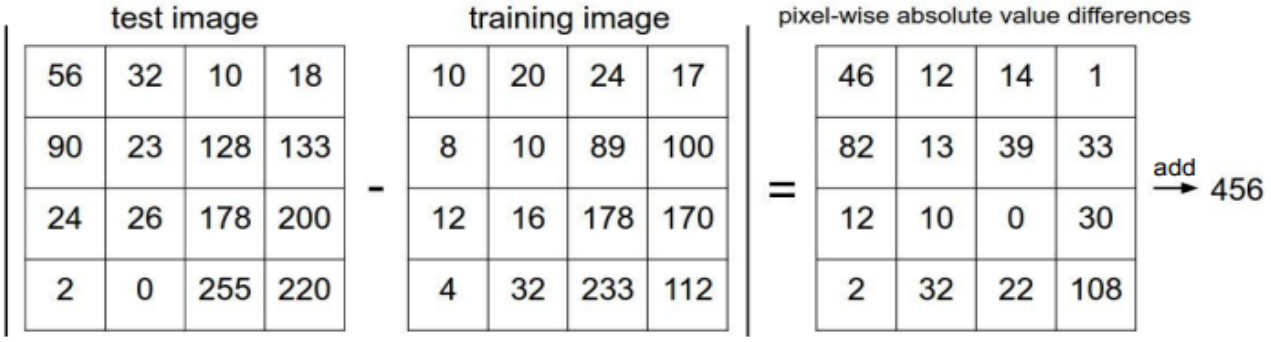
Nearest Neighbor

- Train Step: 모든 학습 데이터 기억
- Predict Step: input data와 train data를 비교해 가장 유사한 이미지로 labeling 예측

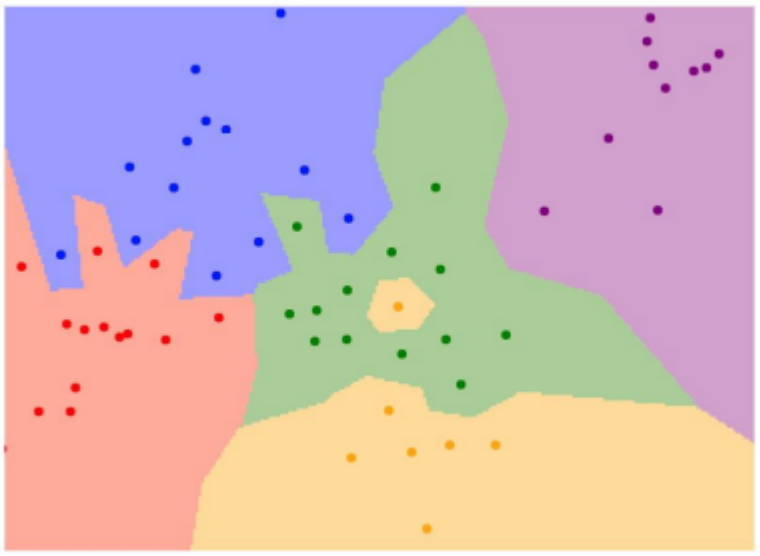
❖ L1 Distance (Manhattan distance)
: 이미지의 각 픽셀을 비교하여 test/train 이미지간의 픽셀 차이 값을 계산한 후 모두 더함.

Distance Metric to compare images

L1 distance: $d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$



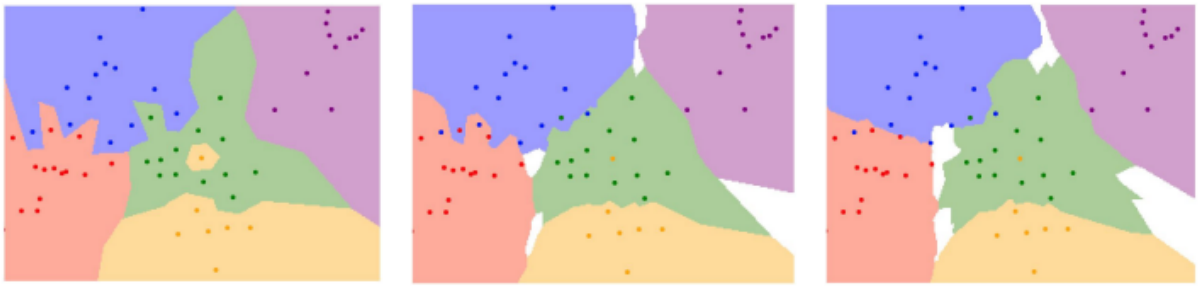
- NN 알고리즘에서 학습시간은 O(1), 테스트시간은 n개의 학습 데이터 전부를 테스트 이미지와 비교해야하므로 O(N)
- train time < test time



nearest neighbor만 보기 때문에 noise/spurious가 생김

K-Nearest Neighbor Classifier

- Distance metric을 이용해서 가까운 이웃을 k개만큼 찾고 neighbor끼리 투표를 한 후 가장 많은 득표수 획득한 레이블로 예측
- k가 커질수록 결정 경계가 부드러워짐

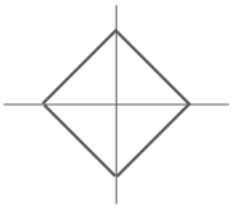


K = 1 K = 3 K = 5

- L2 (Euclidean Distance) : 제곱 합의 제곱근
  - L1 Distance : 특징 벡터의 각각 요소들이 개별적인 의미를 가지고 있는 경우
  - L2 Distance : 특징 벡터가 일반적인 벡터이고, 요소들간의 실질적인 의미를 잘 모르는 경우

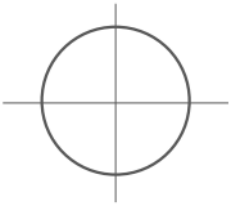
L1 (Manhattan) distance

$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$



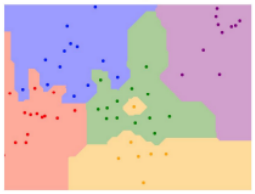
L2 (Euclidean) distance

$$d_2(I_1, I_2) = \sqrt{\sum_p (I_1^p - I_2^p)^2}$$



L1 (Manhattan) distance

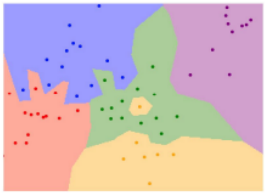
$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$



K = 1

L2 (Euclidean) distance

$$d_2(I_1, I_2) = \sqrt{\sum_p (I_1^p - I_2^p)^2}$$



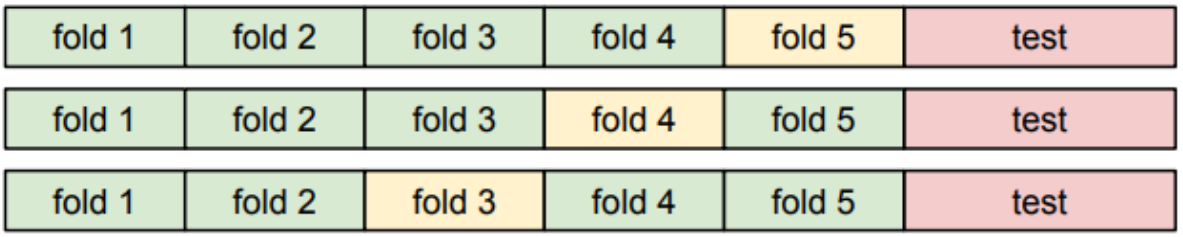
K = 1

- L1 : 결정 경계가 좌표축에 영향을 받음
- L2 : 부드러움

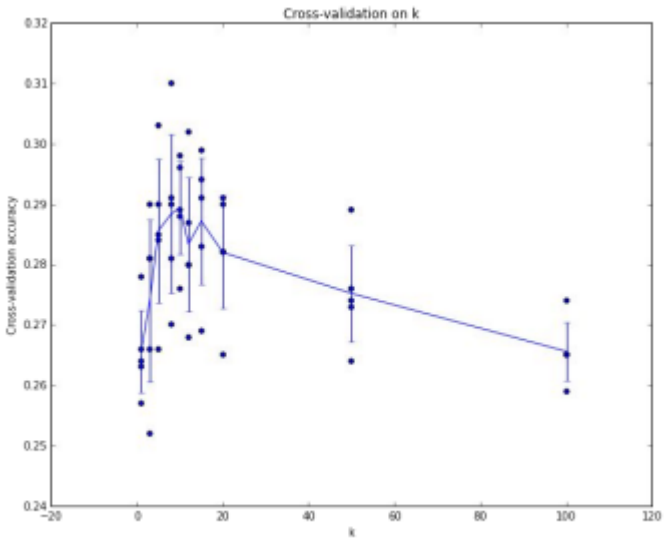
Hyperparameter

- K, 거리척도와 같은 파라미터
- 학습 전 선택
- Hyperparameter 정하는 방법
    a. 데이터를 train set, validation set, test set으로 나눔
    b. 다양한 하이퍼파라미터로 train set 학습
    c. validation set으로 검증 후 가장 좋았던 하이퍼파라미터 선택
    d. 가장 좋았던 classifier를 가지고 test set에서는 한번만 수행

❖ cross validation (교차 검증)



- 우선 테스트 데이터를 정해놓은 후 마지막에 사용
- 나머지 데이터는 train data를 여러부분으로 나눔
- 번갈아가면서 validation set 지정
- 위 예제에선 연두색 fold에서 하이퍼 파라미터를 학습시키고 노란색 fold에서 알고리즘 평가

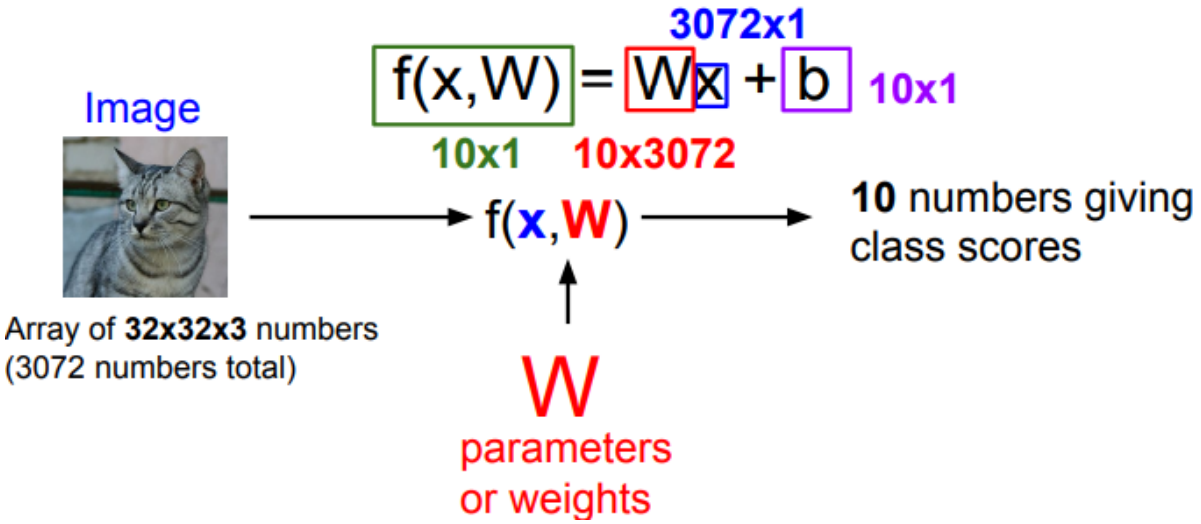


x축 : k-nn의 k값, y축: 분류 정확도
각 k마다 5번의 cross validation을 통해 알고리즘이 얼마나 잘 동작하는지 알려줌
테스트셋이 알고리즘 성능 향상에 미치는 영향을 알아보는 데에 도움이 됨
validation folds 별 성능의 분산을 알 수 있음

Linear Classification

: NN과 CNN의 기반 알고리즘, parametric 모델의 가장 단순한 형태

- ❖ parametric model



x : input image
W: parameters or weights

x와 W를 가지고 10개의 숫자 출력
이 숫자는 CIFAR-10의 각 10개의 카테고리의 스코어
고양이의 스코어가 높다는 건 x가 고양이일 확률이 크다는 것을 의미

- bias term
  : 데이터와 무관하게 특정 클래스에 우선권 부여
  : bias는 데이터와 독립적으로 각 카테고리에 연결됨
  : bias는 각 클래스에 scailing offsets을 더해줌