군집화
비지도학습
데이터 집단을 대표할 수 있는 중심점을 찾는 것

K-means
가깝게 위치하는 데이터를 비슷한 특성을 가진 데이터끼리 묶어 k개의 군집으로 군집화하는 것
<제대로 작동하지 않는 경우>
- 데이터 분포가 특이한 케이스일 경우
- 군집의 밀도가 다를 경우
- 군집의 크기가 다를 경우
<step>
1. 군집 개수(k) 설정
2. 중심점 설정
3. 모든 데이터를 각각 가장 가까운 군집으로 할당
4. 군집의 중심점을 해당 군집에 속하는 데이터들의 중간 지점으로 재설정
5. 데이터를 군집에 재할당

EM
기댓값 최대화 알고리즘 -> 관측되지 않는 잠재변수에 의존하는 모델
<step>
1. 매개변수 θ를 임의의 값으로 초기화
2(E-step). 주어진 매개변수 값에 관한 잠재변수 Z를 추정
3(M-step). Z를 이용하여 θ 다시 추정
4. θ와 Z값이 수렴할 때까지 2, 3단계 반복
=> 해가 수렴하거나 반복 수를 채우면 학습 완성

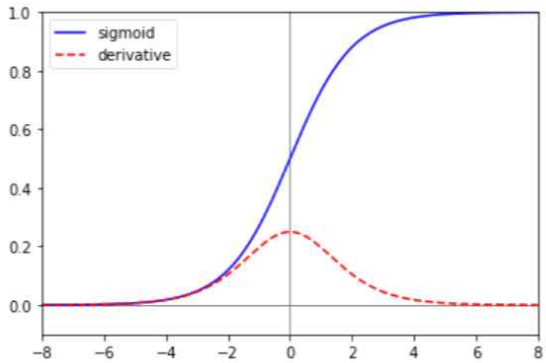
DBSCAN
밀도 기반 클러스터링
- 지정거리(eps) : 군집을 탐색할 거리
- 데이터 개수(n) : 지정거리 내 필요한 최소 데이터 개수
<step>
1. e, n 설정을 통해 밀도 높은 지역 정의
2. 밀도 높은 지역을 만족하는 core point를 찾고 그 지역을 군집으로 할당
3. 해당 지역 안에 core point를 만족하는 데이터가 있다면 그 지역을 포함하여 군집 확장
4. 해당 지역 안에 더 이상 core point를 정의할 수 없을 때까지 2, 3 단계 반복
5. 어떤 군집에도 해당되지 않은 데이터 noise point로 정의

Perceptron
1세대 딥러닝
다수의 입력을 받아 하나의 신호를 출력
단일 층 구조
입력층 노드와 출력층을 연결하는 에지는 가중치 w를 가짐
xor 연산 학습하지 못함(선형 분류로 판단 불가능)
- 임의로 설정된 w로 시작
- 학습 데이터를 입력하여 기댓값이 나오지 않은 경우 w 개선 (=학습)
- 입력 값이 w와 곱해져서 뉴런에 전송
- 뉴런에서는 모든 입력값의 전체 합이 임계치를 넘는지 판단 (넘으면 1, 아니면 0 출력)
  -> 계단함수

모든 학습 데이터를 정확히 분류시킬 때까지 학습이 진행되기 때문에 데이터가 선형적으로 분리될 수 있을 때 적합

Multi-layered Perceptron
2세대 딥러닝
입력과 출력 사이에 하나 이상의 은닉층 추가하여 학습
비선형 분포 데이터에 대해 학습 가능
층과 층 사이에서 선형 방정식을 이용하여 값을 계산한 뒤 넘김 따라서, 선형 데이터를 비선형으로 바꿔주기 위해 활성화 함수 사용

활성화 함수
- 시그모이드 함수
  입력으로 받은 값을 0과 1 사이의 값으로 바꿔줌
  입력으로 받은 값의 절댓값이 크면 기울기가 0으로 수렴
  -> 경사하강법 적용하여 가중치 업데이트할 시 기울기가 0에 가까워져 기울기 소실 현상 발생
  딥러닝에서 잘 사용하지 않음



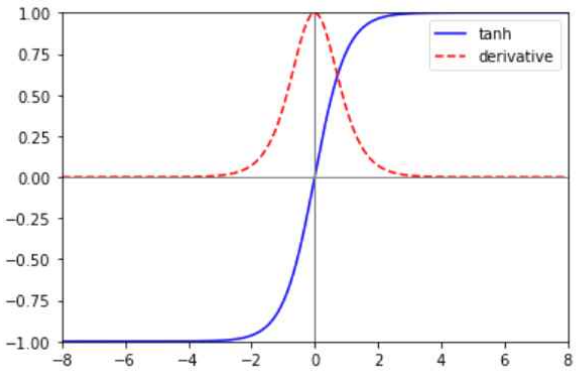
$$sigmoid(x) = \frac{1}{1+e^{-x}}$$

- tanh함수

시그모이드 함수의 문제를 해결하기 위해 나온 함수

선형 함수의 결과를 -1부터 1까지 비선형 형태로 변환

평균이 0이기 때문에 결과값이 양수로 편향되어 있던 문제는 해결되었으나 나머지 문제 해결 X

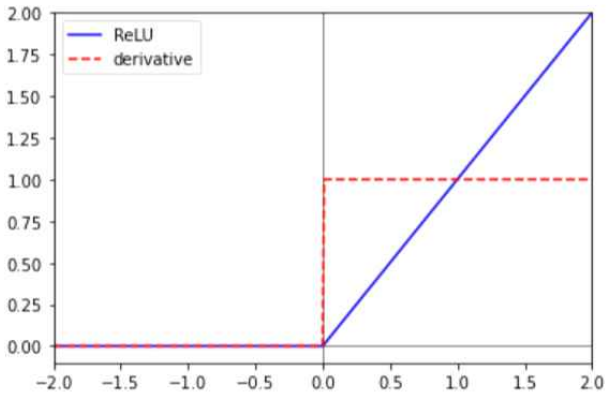


$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

- ReLU 함수

0보다 큰 입력값의 경우 그대로 출력, 0 이하의 값은 다음 층에 전달 X

기울기 소실 문제 완화, CNN에서 자주 쓰임



$$ReLU(x) = \max(0, x)$$

- softmax

모델의 출력값을 0과 1 사이의 값으로 출력

출력 값들의 총합은 항상 1

분류하고 싶은 클래스의 수만큼 출력으로 구성

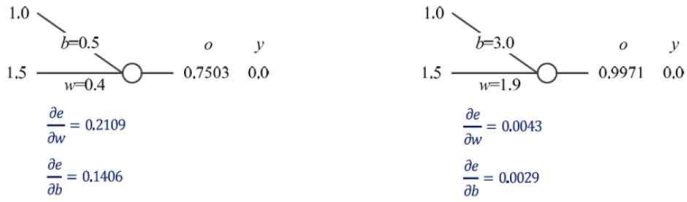
가장 큰 출력값을 부여받은 클래스가 확률이 가장 높음

손실 함수

예측값과 실제값과의 차이를 비교하기 위한 함수

- MSE

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

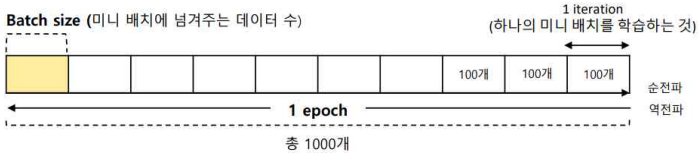


✓ Gradient를 계산해보면 왼쪽의 Gradient가 더 큼
✓ 더 많은 오류를 범한 상황이지만 MSE로 측정할 경우, 더 낮은 벌점을 받게 되는 상황

* 이거 이해 안됨

- 교차 엔트로피
하나의 변수가 서로 다른 분포를 가질 경우 해당 분포들의 차이
- 로그우도 목적함수
활성화함수로 softmax가 사용됐을 경우 목적함수로 사용
정답에 해당하는 노드만 확인
* 식을 다 알아야됨?

딥러닝의 최적화 기법
- 데이터 전처리
정규화 / 표준화 / one-hot encoding
- 가중치 초기화
가장 작은 error가 도출되도록 가중치 설정
- 경사하강법 & 모멘텀
· 경사하강법
기울기가 최소가 되는 지점에 알맞은 가중치 매개변수를 찾아냄
학습률 크기 중요
* 바이어스랑 가중치의 차이?
· 모멘텀
학습 방향이 바로 바뀌지 않고 일정한 방향을 유지하려는 성질
같은 방향의 학습 진행 시 가속을 가지며 더 빠른 학습 기대 가능
- 적응적 학습률
학습률이 너무 작으면 학습에 많은 시간 소요
학습률이 너무 크면 진동할 가능성이 높음
매개변수마다 자신의 상황에 따라 학습률을 조절
SGD, Adagrad, RMSprop, Adam(RMSprop 개선, 일반적으로 사용)
- Epoch
학습 데이터 셋에 포함된 모든 데이터들이 한번씩 모델을 통과한 횟수



· Batch size : 연산 한번에 들어가는 데이터 크기
· mini Batch : 1 Batch size에 해당하는 데이터 셋
· 배치 사이즈가 클 경우 : 한번에 처리해야 되는 데이터의 양이 너무 많음 / 학습 속도 느림

· 배치 사이즈가 작은 경우 : 가중치 업데이트가 자주 발생해 훈련 불안정
- Batch Normalization
데이터가 다양한 분포를 가지더라도 각 배치 별로 평균과 분산을 이용해 정규화하는 과정
* 알고리즘 4개 다 식까지 알아야되는지
- ReLU 활성함수
- Stochastic Pooling
* 이거 했었나..?

규제 기법
과대적합을 피하는 전략
- 가중치 감소
과대적합에서는 가중치 값이 아주 큼
성능을 유지한 상태로 가중치 크기를 낮추는 규제 기법
* 놈 람다 이런 과정 다 알아야되는지?
- 조기 종료
모델이 과적합되기 전 훈련을 멈추는 기법
훈련 중 주기적으로 성능 검증, 성능이 더 좋아지지 않을 시 훈련 멈춤
Epoch 단위로 성능 검증, Batch 단위로 검증하기도 함
· 조기 종료 기준 : 일시적 변동이 아닌 지속적인 정체 도는 하락일 때 종료
- 데이터 증대
큰 훈련 집합을 사용하여 과대적합을 방지하기 위해 주어진 데이터를 인위적으로 증대/증강
* 예시 알아야됨?
- Dropout
일부 노드만 무작위로 골라 학습시키는 기법
학습 중간에 일정 비율의 노드들의 출력을 0으로 만들어 신경망의 출력 계산
· 적용 순서 : 활성화 함수 적용 후 Pooling 이전일 때 적절