

# 딥러닝\_이론

## 05. 머신러닝 기초

- 머신러닝 : 경험 E를 통해(학습:기술이 data를 경험할수록 성능 향상되는 것) P(지표)의 관점에서 T(작업)의 성능 평가
- 학습 단계 : 데이터 수집 → 데이터 전처리 → 가설 정의(가설을 가지고 판독하여 모델의 종류 정함) → 특징 정의 → 목적 함수 정의(목적 함수를 적게하는  $h(x)$  학습)
- 예측 단계 : 데이터(unknown) → 데이터 전처리 → 특징 추출(학습된 모델 이용) → 예측/평가
- 특징 & 패턴
  - 목적 : 분류 → 좋은 특징 : data 분류가 잘됨
  - 유형 : 선형분리 가능 특징, 비선형 분리 가능 특징, 상관관계 높은 특징, 멀티 모달 특징
- 필요성
  - 같은 종류, 그러나 다양한 모양의 데이터가 많음
  - 주어진 데이터로부터 일반적인 규칙을 자동으로 학습
- 구분
  - 지도 학습
    - 기계에게 정답이 무엇인지 알려주면서 학습
    - data label(정답)을 알려주고 그 정답이 맞는지 특징 추출
  - 비지도 학습
    - 정답 x, 데이터 자체의 특성 바탕으로 학습
  - 강화학습 : 기계가 주어진 상태에 대해 최적의 행동 선택/학습데이터 대신 주어진 상태에 맞춘 행동의 결과에 대한 보상으로 성능 향상
  - 준지도학습 | 정답이 있는 데이터, 없는 데이터 모두 학습에 사용
- 지도 학습 : target value dataset이 주어진 데이터
  - 분류 :  $y(\text{target})$ 이 label, 맞으면 1 아니면 0으로 길이 discrete을 구분짓는 것
  - 회귀 :  $y(\text{target})$ 이 연속적인 값(경향), real value인 방법
    - 목적함수(error) 를 최소화하여 회귀 모델을 학습
- 비지도 학습
  - 군집화 : 주어진 데이터 샘플들을 몇 개의 클러스터로 그룹핑
    - 계층적 군집 : 분리형, 응집형 (서로 반대)
    - 포인트 할당 군집 : k-means 군집 (중심값 계속 update하여 sample number들 바꿈)
    - 샘플 간의 거리 함수(d)에 대한 공리
      - $d(x,y)=0$ 이면  $x=y$
      - $d(x,y)=d(y,x)$  거리이므로
      - $d(x,y) \leq d(x,z)+d(y,z)$  (z가 x,y를 이은 직선위에 있다면 둘이 같음)
  - 차원축소
    - 데이터의 특성을 유지하면서도 데이터를 표현하는 차원의 수를 줄임
      - 투영(projection) : 내적을 이용하여 선형 변환
      - 매니폴드 학습 : 비선형 변환
- 강화학습 : 보상을 최대화하기 위해 환경과 상호작용을 하는 agent를 학습 (보상 최적화)

- 데이터 집합 구성
  - 학습 데이터 : 모델을 학습(파라미터 값 결정) 하는 데 사용
  - 검증 데이터 : 학습이 완료된 모델 검증하는데 사용
  - 테스트 데이터 : 최종 모델에 대한 바이어스 없는 평가에 사용
- 모델 설계 요소
  - 파라미터 : 모델 내부에서 결정되는 변수, 데이터로부터 결정(training data)
  - 하이퍼 파라미터 : 모델링할 때 사용자가 직접 결정, 경험 법칙에 의해 결정
    - 학습률, sigma(SVM), k값(knn)
- 교차검증
  - 데이터의 수가 적을 때 사용
- 배치 학습
  - 모든 학습데이터 한꺼번에 학습
  - 일반적으로 오프라인에서 수행
  - 추가로 확보된 데이터를 학습에 사용하려면 처음부터 다시 수행
  - 시간, 자원 많이 소요
- 온라인 학습
  - 데이터를 미니 배치단위로 나누어 순차적으로 학습
  - 학습 단계 빠르고 비용 적음
  - 학습 끝난 데이터 제거 가능 → memory 절약
  - 학습률 설정이 중요 → 높이면, 데이터에 빠르게 적응 / 낮추면, 새로운 데이터에 덜 민감
- Underfitting & Overfitting
  - high bias (편향) : 너무 적은 수의 특징을 학습
  - high variance (분산) : 너무 많은 수의 특징을 학습
  - 모델의 복잡도가 낮을수록 편향이 커 underfitting될 확률 높아짐
  - 모델의 복잡도가 높을수록 학습 데이터에 overfitting될 확률 높아짐
- overfitting 해결
  - 데이터 증식 : 더 많은 데이터 수집, 데이터 변형 noise 추가
  - 모델의 복잡도 감소 : parameter 수 감소, hidden\_layer 수 감소
  - dropout(신경망 학습시) : 신경망의 일부 사용하지 않음
  - 조기 종료
  - 정규화
    - L2 정규화 : loss값 더 크게 하여 penalty를 줌  
→ 목적함수, gradient descent(기울기 감소)  
→ weight decay : 특정 가중치가 비정상적으로 커져서 학습에 영향을 주는 것을 방지

- 
- 딥러닝
    - 머신러닝 알고리즘 중 선형 또는 비선형 transform 함수로 구성된 processing layer를 다층으로 갖고 있는 deep graph를 사용하여 입력으로부터 고차원적인 추상화를 목적으로 하는 것들의 집합
    - 구조
      - Feed-Forward : 순방향
      - Feed-Back : 역방향

- Bi-Directional : 양방향
- Recurrent : 순환신경망 (시계열에 사용)
- 주목받지 못했던 이유
  - 성능 측면
    - layer 개수가 적으면 좋은 성능을 내기 어려움
    - layer 개수가 많을 시, 심층부 layer까지 잘 학습시키는 학습 알고리즘의 부재
- rectified linear unit (ReLU)와 같은 새로운 activation 함수의 적용
- dropout과 같은 새로운 학습법
- practical한 측면
  - overfitting을 피하기 위해서 많은 수의 labeled data 필요
  - 학습해야 하는 parameter 가 너무 많아 학습에 걸리는 시간 너무 많음
- crowdsourcing 플랫폼 출현

## 퍼셉트론

: 기본적인 퍼셉트론의 경우, 초평면으로 n차원 공간을 두 개의 결정 영역 구분

- 기본 학습
  - (-0.5,0.5) 범위에서 초기 가중치를 임의로 할당
  - 정답이 있는 학습 샘플에 적용
  - p번째 반복(퍼셉트론에 주어진 p번째 학습 샘플)
  - $e(p)$  오차 =  $Y_d(p)$  목표 출력 -  $Y(p)$  실제 출력
- 퍼셉트론 학습 규칙
  - 오차  $e(p)$ 가 양일 때, 퍼셉트론의 출력  $Y(p)$ 를 증가
- 각 퍼셉트 입력에 대해  $x_i(p)$  주어진 데이터 X  $w_i(p)$  값이 총합  $X(p)$ 에 기여
  - $x_i(p)$ 가 양일 때, 가중치  $w_i(p)$ 가 커지면  $Y(p)$ 도 커짐
- 가중치 갱신
  - $w_i(p+1) = w_i(p) + a \times x_i(p) \times e(p)$
  - $e(p) > 0, x_i(p) > 0$   $w_i(p+1)$ 도 증가
  - 학습률 : 1보다 작은 양의 실수
- 역전파 학습 알고리즘 (정리)
  1. 초기화 : 가중치 초기화
  2. 활성화
  3. 가중치 학습
  4. 반복
- 역전파 학습의 수렴 가속
  - 작은 학습률  $a$  : 가중치에 작은 변화, 매끄러운 학습 곡선
  - 큰 학습률  $a$  : 가중치가 크게 변하므로 빠른 수렴
  - 휴리스틱을 이용해 수렴 가속
    - 오차 제곱의 합의 변화량이 몇번에 EPOCH에 대해 계속 같은 부호라면? 학습률 증가시킴
    - 계속 엇갈리면 학습률 매개변수  $a$ 값을 감소