Chapter 4

어경빈

4.1 머신러닝의 4가지 분류

- 4.1.1 지도 학습
- 4.1.2 비지도 학습
- 4.1.3 자기지도 학습
- 4.1.4 강화학습

4.2 머신러닝 모델평가

- 4.2.1 훈련, 검증, 테스트 세트
- 훈련, 테스트로만 해도 되지 않나? -> 검증 세트도 따로 있어야만 오버피팅 방지에 효과적이다. 꼭 back propagation을 통한 훈련이 아니더라도, dataset에 맞게 hyper-parameter를 조정하는 것도 간접적으로 overfitting에 일조할 수 있기 때문이다.

4.2 머신러닝 모델평가

- 검증 세트를 만드는 방법은 크게 3가지다.

- 홀드검증
  - training set에서 검증세트를 따로 빼놓는 것.

- k겹 교차검증
  - k개의 블록을 나눠 놓고 매번 검증으로 사용하는 블록을 다르게 잡는 것.

- 반복 k겹 교차검증
  - k개의 training할 블록들을 shuffle을 하는 것.

4.3 데이터 전처리, 특성공학, 특성학습

- 4.3.1 데이터 전처리

- 벡터화 : 입력받은 데이터는 모두 tensor의 형태로 변환해 주어야 한다.(문자 데이터라도 vector화 해야 함.)

- 값 정규화 : 0 ~ 1 사이의 값, 그게 힘들다면 평균이 0이고 표준편차가 1인 분포로 정규화를 해주어야 함.

- 누락된 값 다루기 : 종종 어떤 일부 특성이 0인 값은 있어도 되지만, training엔 없다가 test set에만 있다면 안 된다.

4.3 데이터 전처리, 특성공학, 특성학습

- 4.3.2 특성공학
- 아날로그 시계의 경우 3가지 형태로 데이터를 넣어줄 수 있다.
- 1. 아날로그 시계의 이미지
- 2. 시계의 시분초 침의 끝의 좌표
- 3. 시계의 시분초 침의 각도
- 딥러닝은 별로 고려해 줄 필요가 없다.

4.4 과대적합과 과소적합(방지)

- 4.4.1 네트워크 크기 축소
- 유닛의 수가 너무 많으면 오버피팅이 일어날 확률이 몹시 큼.
- 반대로 너무 적으면 언더피팅이 일어날 확률이 몹시 큼.
- 적절한 수는 오직 경험적으로만…..

4.4 과대적합과 과소적합(방지)

- 4.4.2 가중치 규제 추가
- 첨부된 링크에 설명이 기똥차게 되어있다.(https://light-tree.tistory.com/125)

4.4 과대적합과 과소적합(방지)

- 4.4.3 드롭아웃
  - 딥러닝을 할 때에, 매 Layer를 지날 때마다 출력값을 정해진 비율에 따라 무작위로 선택하여 0으로 만든다.
  - 무작위로 0을 만들기 때문에 유닛들과의 training set에 대한 '부정한 협업'을 깨버릴 수 있다.

4.5 보편적인 머신러닝 작업흐름

- 4.5.1 문제정의와 데이터셋 수집
  - 어떤 문제를 해결할지(출력값)정하고, 딥러닝 학습을 위한 데이터 셋 수집.

- 4.5.2 성공 지표 선택
  - 보통 Accuracy(정확도)를 지표로 선택

- 4.5.3 평가 방법 선택
  - 검증방법이 더 맞는듯. 대부분의 딥러닝의 경우 데이터셋이 풍부하여 홀드검증으로도 충분하다고 함.

- 4.5.4 데이터 준비
  - 데이터의 전처리 단계

4.5 보편적인 머신러닝 작업흐름

- 4.5.5 기본보다 나은 모델 훈련하기
  - 통계적 검정력정도의 performance를 내는 것이 목표

- 4.5.6 몸집 키우기 : 과대적합 모델 구축
  - 모델의 유닛들을 추가 및 제거해서 오버피팅과 언더피팅 사이의 적당한 network 사이즈를 정하는 것.

- 4.5.7 모델 규제와 하이퍼 파라미터 튜닝
  - 앞서 했던 다양한 규제방식을 적용해보고, hyper-parameter를 조절하는 단계이다. 대부분의 시간을 소모하는 일.