Keras 4장 – 머신 러닝의 기본 요소

1. 머신 러닝의 네 가지 분류

1) 지도 학습(Supervised Learning): sample이 주어지면 이미 알고있는 target에 매핑하는 방식 ex) binary & multiclass classification, scalar regression

2) 비지도 학습(Unsupervised Learning): target 사용하지 않고 input의 변환을 찾음 ex) 차원 축소(dimensionality reduction), 군집(clustering)

3) 자기 지도 학습(Self-supervised Learning): 학습 과정에 사람이 개입하지 않는 지도 학습 ex) 오토인코더

4) 강화 학습(Reinforcement Learning): 환경에 대한 정보를 받아 보상을 최대화하는 행동을 선택 ex) 딥마인드의 Atari 게임 (보상: 게임 점수 최대화)

\* 용어

1. 샘플(sample), 입력(input): 데이터 포인트 1개
2. 예측(prediction), 출력(output): 모델로부터 나오는 값
3. 타깃(target): 정답
4. 예측 오차(prediction error), 손실 값(loss value): 예측과 target 사이의 거리
5. 클래스(classes): 선택 가능한 레이블의 집합 ex) 강아지, 고양이
6. 레이블(label): 분류 문제에서 클래스 할당의 구체적인 예
7. 참 값(ground-truth), 꼬리표(annotation): dataset에 대한 모든 target(일반적으로 사람에 의해 수집)
8. 이진 분류(binary classification)
9. 다중 분류(multiclass classification)
10. 다중 레이블 분류(multilabel classification): 샘플이 여러 개의 레이블로 분류될 수 있는 경우 ex) 한 사진에 강아지, 고양이 모두 있는 경우
11. 스칼라 회귀(scalar regression) ex) boston housing
12. 벡터 회귀(vector regression): 스칼라 회귀처럼 연속된 값의 집합인데, 스칼라가 아닌 벡터임
13. 미니 배치(mini-batch), 배치(batch): 동시에 처리되는 샘플 묶음

2. 머신 러닝 모델 평가

- 목표: overfitting되지 않는 일반화된 모델을 얻는 것

- parameter: network의 weight를 뜻함

- hyper-parameter: layer의 수, unit 수 등의 변수들을 뜻함

- 튜닝: hyper-parameter를 조정하는 행위

- 튜닝을 많이 할 수록 정보 누설(information leak)이 심해지고 이것은 overfitting으로 이어짐

1. 단순 홀드아웃 검증(Simple Hold-Out Validation)
2. K-겹 교차 검증(K-Fold Cross-Validation)
3. 셔플링을 사용한 반복 K-겹 교차 검증(Iterated K-Fold Validation with Shuffling)

\*\* 기억해야 할 것

1. 대표성 있는 데이터: 셔플하지 않으면 train set에 0~7만 담기고 test set엔 8~9만 담기는 현상 발생 -> 셔플해야 함!
2. 시간의 방향: 시계열 데이터에서는 무작위로 섞으면 안 됨
3. 데이터 중복: 같은 데이터 중복되면 overfitting의 원인

3. 데이터 전처리, 특성 공학, 특성 학습

1) 신경망을 위한 데이터 전처리(Data Processing for Neural Networks)

(1) 벡터화(vectorization): input, target을 tensor화 하는 것

(2) 값 정규화(value normarlization): 제각각 다른 특성들을 균일하게(0~1 등) 가공하는 것

(3) 누락된 값 다루기(handling missing values): 0 같은 값이 의미있는 값이면 조심해서 사용 (?, # 등으로 대신 사용하기!)

2) 특성 공학(Featured Engineering)

- 학습이 아닌 하드코딩을 통해 알고리즘을 적용

- 특성을 더 간단한 방식으로 표현

- 과거에는 중요했으나 현재는 자동 추출(automatic extraction)로 필요하진 않음

- 그러나 2가지 이유로 알아야 함

(1) 좋은 특성은 적은 자원을 사용하여 더 멋지게 문제 해결 가능

(2) 가용한 샘플의 개수가 적으면 특성 정보가 매우 중요

4. 과대적합과 과소적합(Overfitting and Underfitting)

드롭아웃: 불필요한 우연한 규칙을 깨트려서 overfitting을 방지

(계속)