Machine Learning & Deep Learning

Lecture 22

Keras를 이용한 딥러닝 코드 분석

과적합(overfitting) 해결을 위해
K겹 교차 검증 방식(K-fold Cross Validation)이 적용된
광석 vs. 돌 분류

김희숙 (H.S.Kim)

광석과 돌 분류 예제

- 1988년 존스홉킨스대학교의 세츠노스키(Sejnowski) 교수는 2년 전 제프리 힌튼(Geoffrey Hinton) 교수가 발표한 **오차 역전파(Error Backpropagation)** 알고리즘에 관심을 가지고 있었음
- 그는 **은닉층**과 **오차 역전파**가 얼마나 큰 효과가 있는지를 직접 실험해 보고 싶었음
- 오차 역전파 알고리즘을 사용한 신경망이 **광석과 돌을 구분하는 데 얼마나 효과적인지 알아보기 위해서** 광석과 일반 돌을 가져다 놓고 **음파탐지기(sonar)를 쏜 후 그 결과를 데이터로 정리함** → sonar.csv

광석과 돌을 분류하기 위한 데이터 → sonar.csv

- 208개로 구성된 샘플
- 60개의 속성과 1개의 클래스로 구성
- 모든 컬럼이 실수형(float64)인데, 맨 마지막 컬럼만 R(Rock) 또는 M(Minerals) 객체형인 것으로 보아 마지막에 나오는 컬럼은 클래스이며 데이터형 변환이 필요한 상태임을 알 수 있음

Range Index: 208 entries,0 to 207			
Data columns (total 61 columns):			
0	208	non-null	float64
1	208	non-null	float64
2	208	non-null	float64
3	208	non-null	float64
4	208	non-null	float64

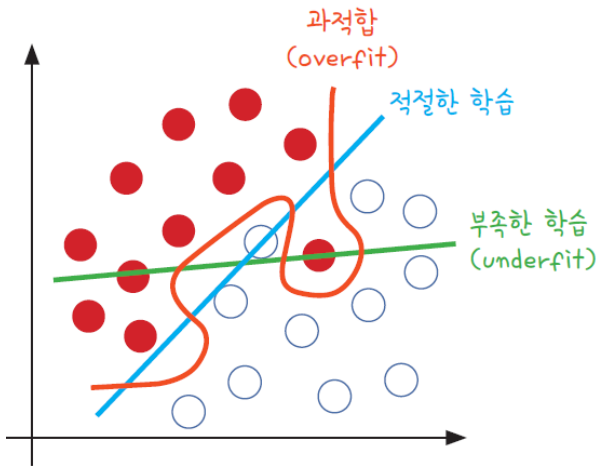
	0	1	2	3	...	59	60
0	0.02	0.0371	0.0428	0.0207	...	0.0032	R
1	0.0453	0.0523	0.0843	0.0689	...	0.0044	R
2	0.0262	0.0582	0.1099	0.1083	...	0.0078	R
3	0.01	0.0171	0.0623	0.0205	...	0.0117	R
4	0.0762	0.0666	0.0481	0.0394	...	0.0094	R

광석과 돌 분류 예제

- 세즈노스키(Sejnowski) 교수는 **데이터셋**을 사용하여 오차율을 낮추면서 정확도를 높이는 딥러닝을 수행했는데 완전히 **새로운 데이터**에 적용하면 **광석과 돌에 해당하는 2개의 그룹으로 정확히 분류되지 않음**을 발견 → 이유 : **과적합(overfitting) 때문**

- **과적합 발생하는 이유 : 층이 너무 많거나 변수가 복잡해서 발생**하기도 하고, **학습 데이터셋과 테스트 데이터셋을 구분하지 않고 딥러닝을 수행할 경우에 발생**하기도 함

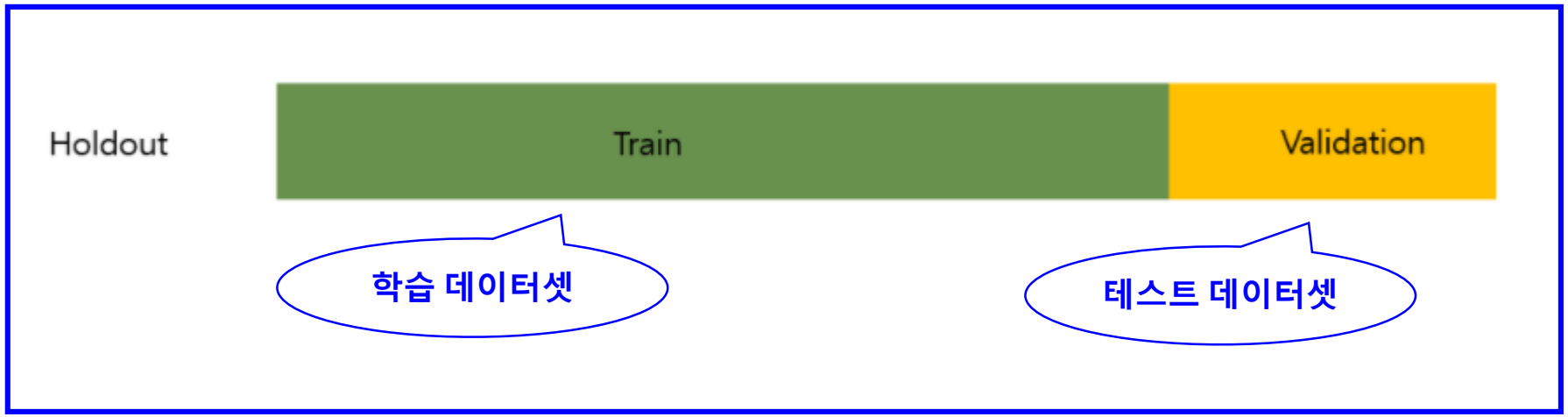
- 딥러닝은 실행 단계에서 입력층, 은닉층, 출력층의 노드들에 상당히 많은 변수들이 투입됨 → 딥러닝을 진행하는 동안 **과적합에 빠지지 않게 늘 주의해야 함**



과적합이 일어난 경우(빨간색) vs. 학습이 제대로 이루어지지 않은 경우(초록색)

과적합(overfitting)을 방지하기 위해서

- **홀드 아웃 교차 검증(Holdout Cross Validation) 방식**
  - ✓ 일반적으로 사용하는 데이터셋을 **학습 데이터셋**과 validation에 해당하는 **테스트 데이터셋**으로 나누어 사용하는 것
  - ✓ 보통 데이터셋을 7 : 3 / 8 : 2 / 9 : 1 등의 비율로 나누어 사용

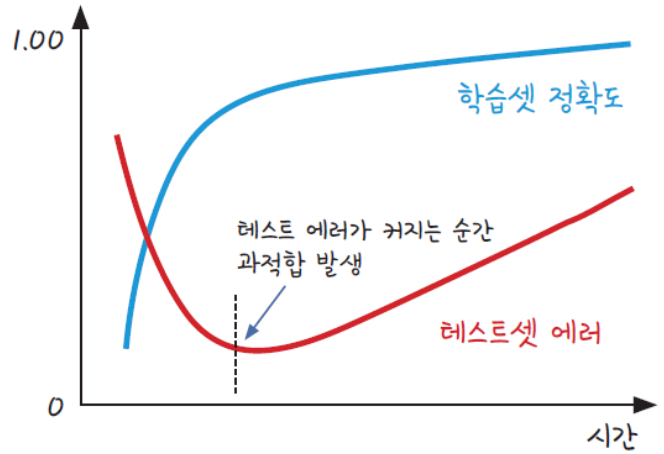


광석과 돌 분류 예제

- **앞에서 다른 예제들을 통해서 실습했던 딥러닝 방식은 과적합을 방지하기 위해서**
  - ✓ 딥러닝을 위해서 전체 데이터셋의 일부는 **학습 데이터셋**으로 사용하고 나머지 30%정도의 데이터셋을 **테스트 데이터셋**으로 준비했다.
  - ✓ 준비한 30% 정도의 **테스트 데이터셋**을 가지고 실행되었된 딥러닝 모델을 기반으로 딥러닝 모델을 테스트한 후 정확도(accuracy)를 계산하였다.
  - ✓ 이 방법은 빠른 시간에 모델 성능을 파악하고 수정할 수 있도록 도와주고 있다.
  - ✓ 머신러닝의 최종 목적은 과거의 데이터를 대상으로 학습되었던 딥러닝 학습 모델을 기반으로 새로운 데이터를 예측하는 것
  - ✓ 향후 예측을 위해서 학습에 사용되지 않은 **테스트 데이터셋**을 만들어 정확한 평가를 병행하는 것이 매우 중요함

과적합(overfitting) 확인 방법

- **학습 데이터셋만 가지고 딥러닝을 실행할 때 층을 더하거나,**
  에포크(epoch) 값을 높여 실행 횟수를 늘리면 정확도가 계속해서 올라갈 수 있음
  - ✓ 이 경우 학습이 깊어져서 학습 데이터셋 내부에서의 성공률은 높아져도 테스트 데이터셋에서는 효과가 없다면 **과적합이 일어나고 있는 것**



학습이 계속되면 학습 데이터셋에서의 정확도는 계속 올라가지만 테스트 데이터셋에서는 과적합이 발생

홀드 아웃 교차 검증(Holdout Cross Validation) 사용시

- 아래의 그림은 **초음파 광물 예측 데이터**를 만든 세츠노스키(Sejnowski) 교수가 실험 결과를 발표한 논문의 일부
- **은닉층의 수가 커짐에 따라 학습 데이터셋의 예측율**과 **테스트 데이터셋의 예측율**이 어떻게 변하는지를 보여줌
  - ✓ 학습량이 늘어날수록 **학습 데이터셋을 통한 예측율은 계속해서 올라가지만,**
    테스트 데이터셋을 이용한 예측률은 오히려 떨어지는 것을 확인할 수 있음
  - ✓ 학습을 진행해도 **테스트 결과가 더 이상 좋아지지 않는다면 그 지점에서 학습을 멈춰야 함**
  - ✓ 이때의 학습 정도가 가장 적절한 것으로 볼 수 있음

TABLE 2
Aspect-Angle Dependent Series

Number of Hidden Units	Average Performance on Training Sets (%)	Standard Deviation on Training Sets (%)	Average Performance on Testing Sets (%)	Standard Deviation on Testing Sets (%)
0	79.3	3.4	73.1	4.8
2	96.2	2.2	85.7	6.3
3	98.1	1.5	87.6	3.0
6	99.4	0.9	89.3	2.4
12	99.8	0.6	90.4	1.8
24	100.0	0.0	89.2	1.4

Summary of the results of the aspect-angle dependent series of experiments with training and testing sets selected to include all target aspect angles. The standard deviation shown is across networks with different initial conditions.

학습 데이터셋과 **테스트 데이터셋** 정확도 측정의 예(RP Gorman et.al.,1998)

광석과 돌 분류 예제

광석과 돌을 분류하기 위한 데이터 → sonar.csv

22_(9 page) 강의용 Sonar (Holdout Cross).ipynb

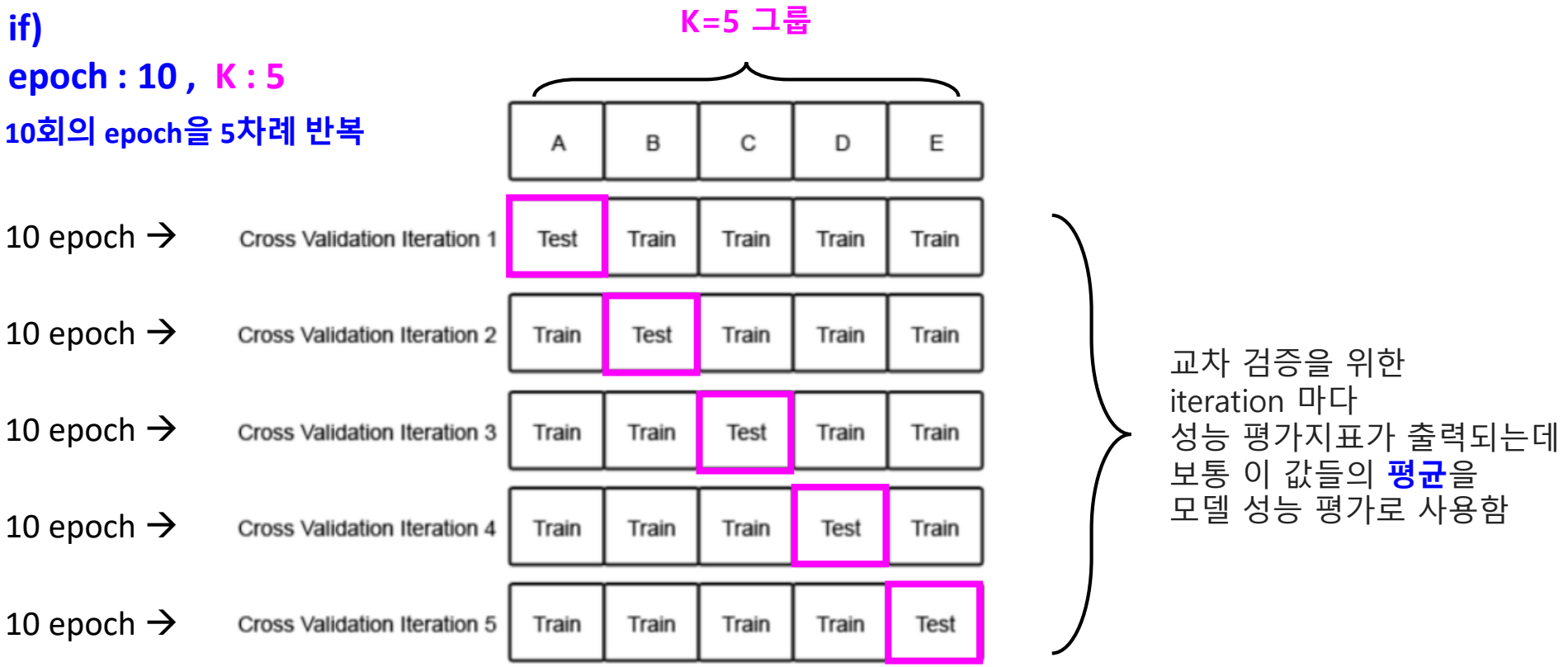
광석과 돌 분류 예제

- **테스트 데이터셋**을 더 정확하게 설정할수록 더 잘 작동한다.
- 하지만 문제는 데이터가 충분하지 않을 경우 발생한다.
- 딥러닝을 수행할 때 어려운 문제 중 하나는 알고리즘을 충분히 테스트하였더라도 데이터가 충분하지 않으면 좋은 결과를 내기가 어렵다.
- 홀드 아웃 교차 검증(Holdout Cross Validation)의 경우, 데이터의 약 70%를 학습 데이터셋으로 사용하고 테스트 데이터셋은 겨우 30%에 불과했다.
- 이 정도 테스트만으로는 실제로 얼마나 잘 작동하는지 확인하기 어렵다.
- **데이터셋의 크기가 작은 경우 테스트 데이터셋에 대한 성능 평가의 신뢰성이 떨어지게 된다.** 만약 **테스트 데이터셋**을 어떻게 잡느냐에 따라 성능이 다르면 우연의 효과로 인해 **모델 평가 지표에 편향이 생기게 된다.**
- 그래서 **홀드 아웃 교차 검증(Holdout Cross Validation)의 단점을 보완**하기 위한 새로운 방법이 등장했다. → **K-겹 교차 검증(K-fold Cross Validation)**

과적합(overfitting)을 방지하기 위해서

- **K-겹 교차 검증(K-fold Cross Validation)**
  - ✓ **모든 데이터가 최소 한 번은 테스트 데이터셋으로 쓰이도록 한다.**
  - ✓ 데이터를 K개의 그룹으로 나누어 하나씩 **교차**시켜가면서 **테스트 데이터셋**으로 사용한다.

if)
epoch : 10 , K : 5
10회의 epoch을 5차례 반복



▪ K-겹 교차 검증(K-fold Cross Validation)

✓ 장점 : **모든 데이터**를 학습 데이터셋과 validation을 위한 테스트 데이터셋에 사용할 수 있고, 이는 과적합(overfitting)의 문제점을 해결하는데 도움이 된다.

✓ 단점 : 딥러닝에 소요되는 시간이 다소 오래걸린다.

광석과 돌을 분류하기 위한 데이터 → **sonar.csv**

22_(13 page) 강의용 Sonar (K-fold Cross Validation).ipynb