인공지능 기초

인공지능_ Day03

김새봄

2023. 06. 28

CONTENTS

NAVER Cloud x BIT BITCamp

회귀분석과 분류분석

회귀분석과 분류분석

- 회귀분석(Regression)

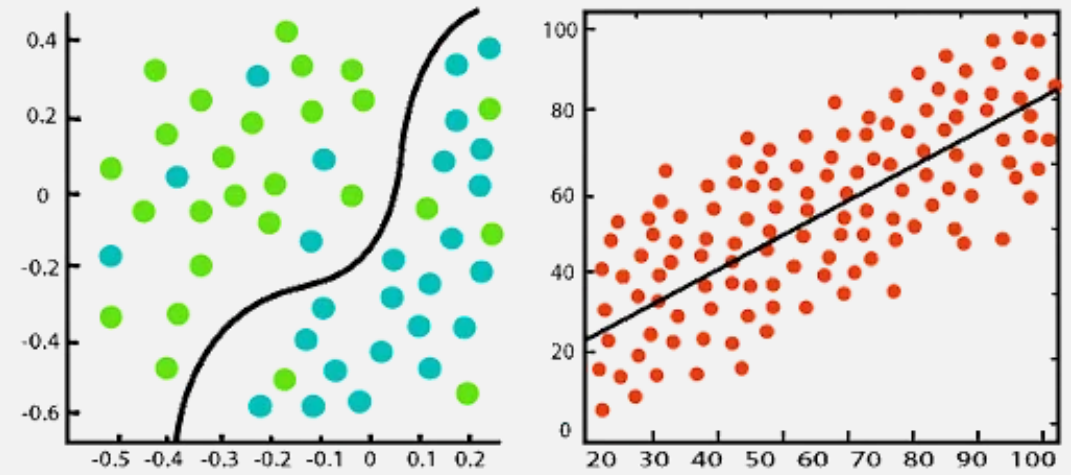
  연속된 값을 예측

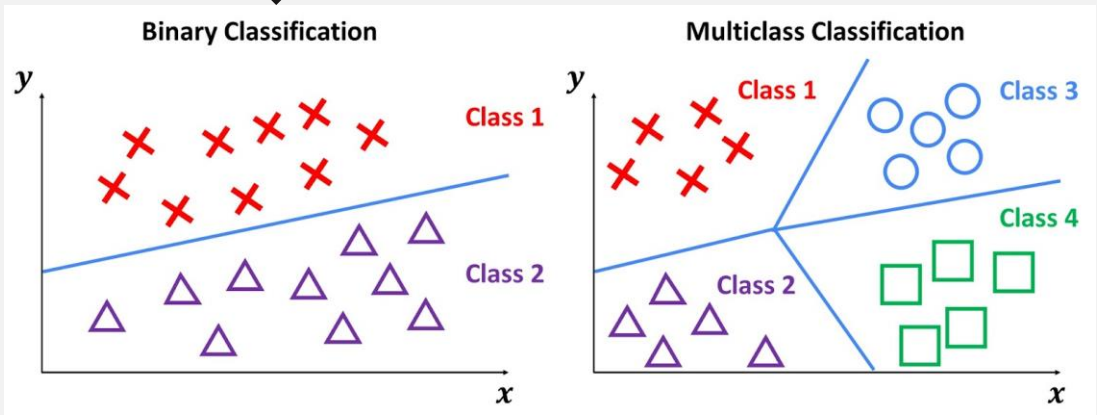
- 분류분석(Classification)

  종류를 예측

  이진 분류(binary classification) 와

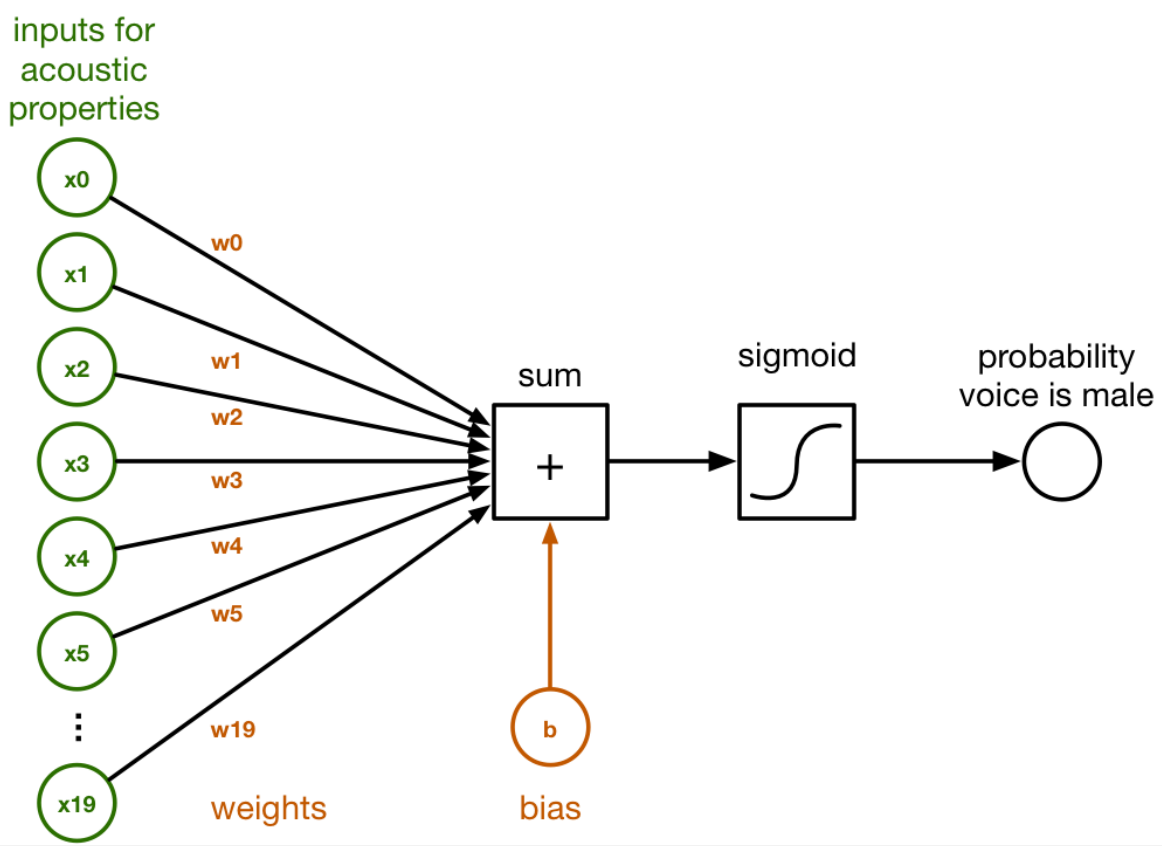
  다중 분류(multi classification)





이미지 출처: https://www.simplilearn.com/regression-vs-classification-in-machine-learning-article
https://datahacker.rs/008-machine-learning-multiclass-classification-and-softmax-function

이진분류

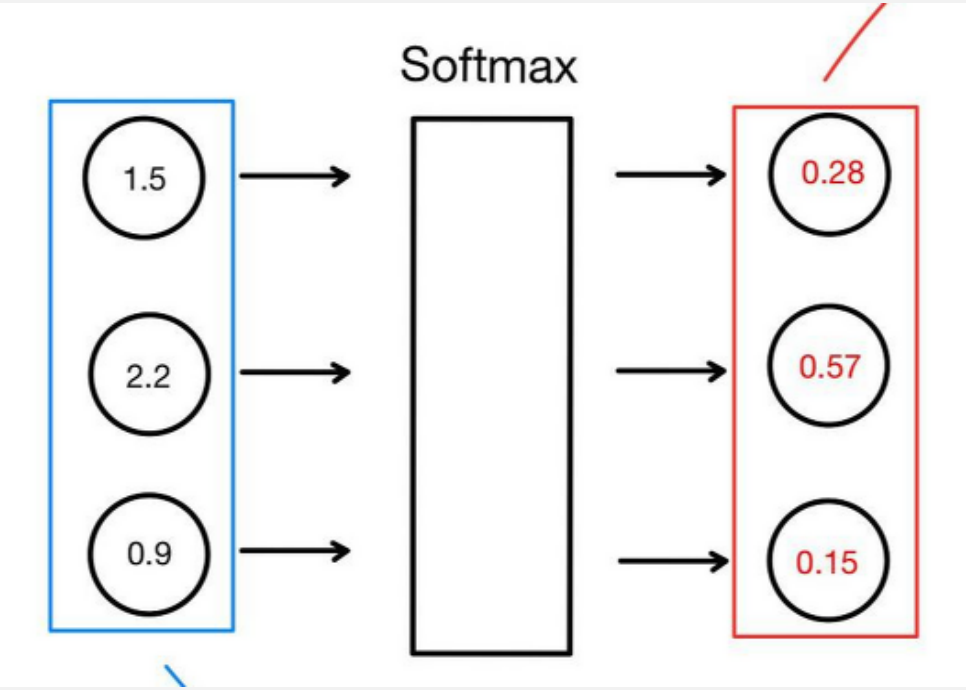


출처 https://wikidocs.net/41256

Figure 2. 이진분류 모델

- 이진분류의 경우 참(True) 또는 거짓(False)을 판별하기 때문에 출력 값이 하나
- 출력 값을 sigmoid 함수를 이용하여 0과 1로 가공
- 로지스틱 회귀(Logistic Regression)는 이진분류모델을 분석하기 좋은 모델
  - 직선보다 적절한 곡선을 통해 분류를 함

다중분류



이미지 출처: https://yhyun225.tistory.com/14

Figure 3. 다중분류 모델의 Softmax 함수

- 타깃의 종류가 여러 개이기 때문에 출력 값도 여러 개
- softmax 함수를 사용하여 0과 1사이의 값으로 가공
- one-hot encoding이라는 기법을 사용

softmax 함수: 각 클래스가 정답일 확률을 표현하도록 0에서부터 1사이의 값으로 정규화(Normalization)를 해주는 함수

***정규화는 왜 해줄까??

- 큰 값을 갖는 변수에 편향되지 않을 수 있음
- X값을 특정 범위로 제한하기 때문에 최적화를 더 빠르게 만든다 (학습 속도 향상)

다중분류

ID	과일
1	사과
2	바나나
3	체리

one-hot encoding

ID	사과	바나나	체리
1	1	0	0
2	0	1	0
3	0	0	1

Figure 4. one-hot encoding

- 타깃의 종류가 여러 개이기 때문에 출력 값도 여러 개
- softmax 함수를 사용하여 0과 1사이의 값으로 가공
- one-hot encoding이라는 기법을 사용

One-hot encoding: 단 하나의 값만 True(1)이고 나머지는 False(0)으로 인코딩하는 것

*** 왜 해줄까??

- 데이터 형태가 0, 1로 이루어졌기 때문에 컴퓨터가 인식하고 학습하기에 용이함
- 각 클래스 간의 차이를 균등하게 취급하도록 함 (즉, 큰 값을 가진 클래스가 중요하다고 인식하지 않도록 함)

optimizer

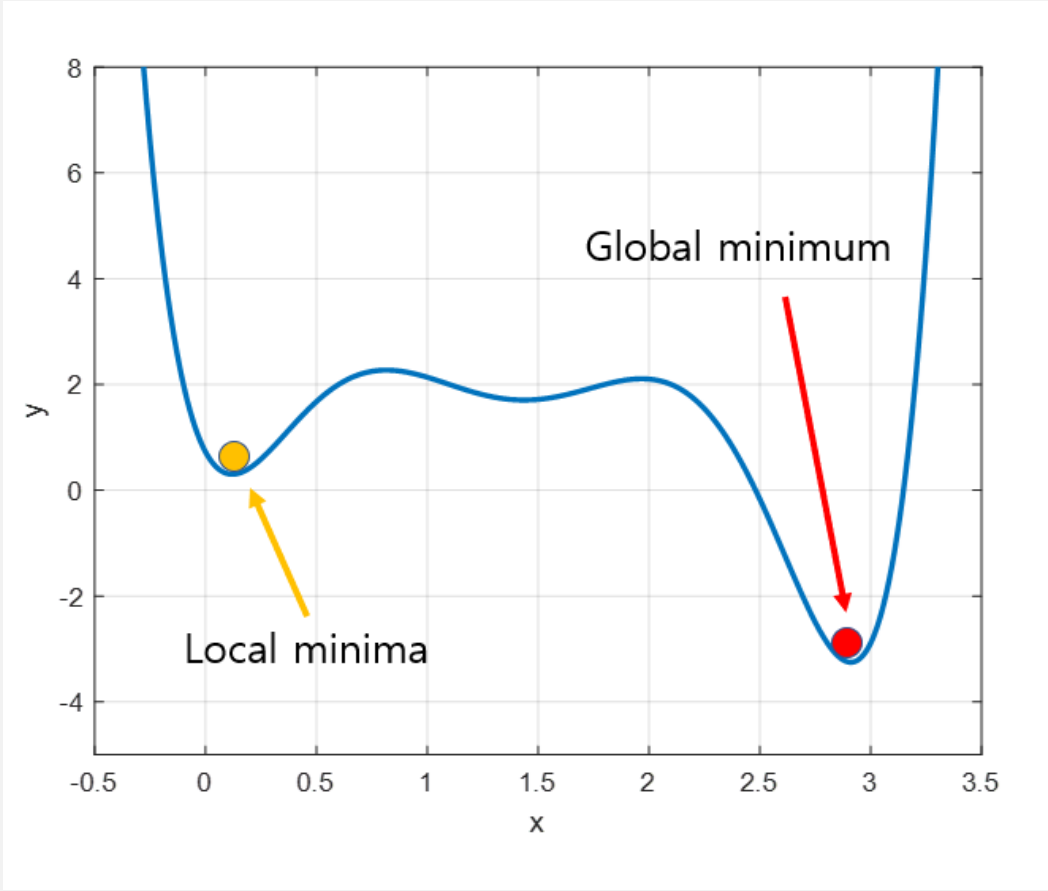
Gradient Descent

함수의 기울기(즉, gradient)를 이용해
x의 값을 어디로 옮겼을 때 함수가 최소값을 찾는지
알아보는 방법

➡ **문제점**

1. 극소값(local minimum)에 도달할 수 있다는 것이
증명되었으나 **전체 최소값(global minimum)**
으로 갈 수 있다는 보장은 없음

2. 훈련이 느림



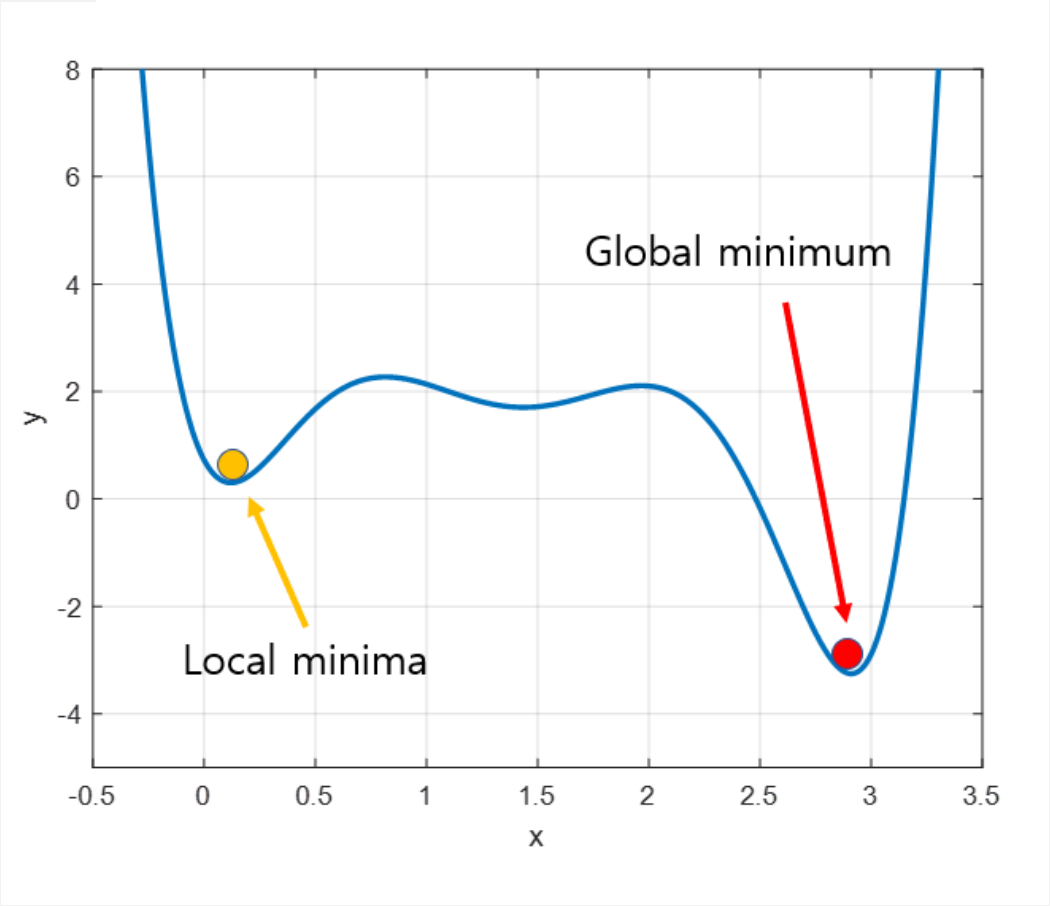
출처 https://angeloyeo.github.io/2020/08/16/gradient_descent.html

Momentum

GD의 문제를 해결하기 위해 보편적으로 사용되는 방법

관성(momentum)을 적용하여 변수가 가던 방향으로
계속 가도록 속도(velocity)를 추가한 것

global minima에 이르기 전에
기울기가 0이 되는 local minima에 빠지는 것을 방지



출처 https://angeloyeo.github.io/2020/08/16/gradient_descent.html

AdaGrad

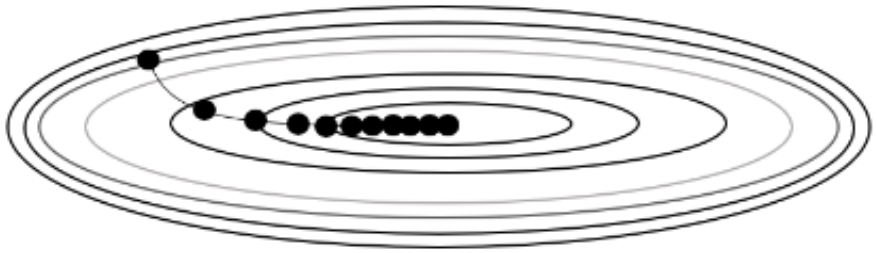
Momentum은 gradient의 방향에 초점을 두었다면,

AdaGrad는 step size에 초점

이미 학습이 많이 된 변수는 학습을 느리게, 학습이 아직 덜 된 드물게 등장한 변수라면 학습을 더 빨리 훈련함

➡ **문제점**

학습이 오래 진행이 될 경우 step size가 너무 작아져서 거의 움직이지 않는 상태가 됨



AdaGrad

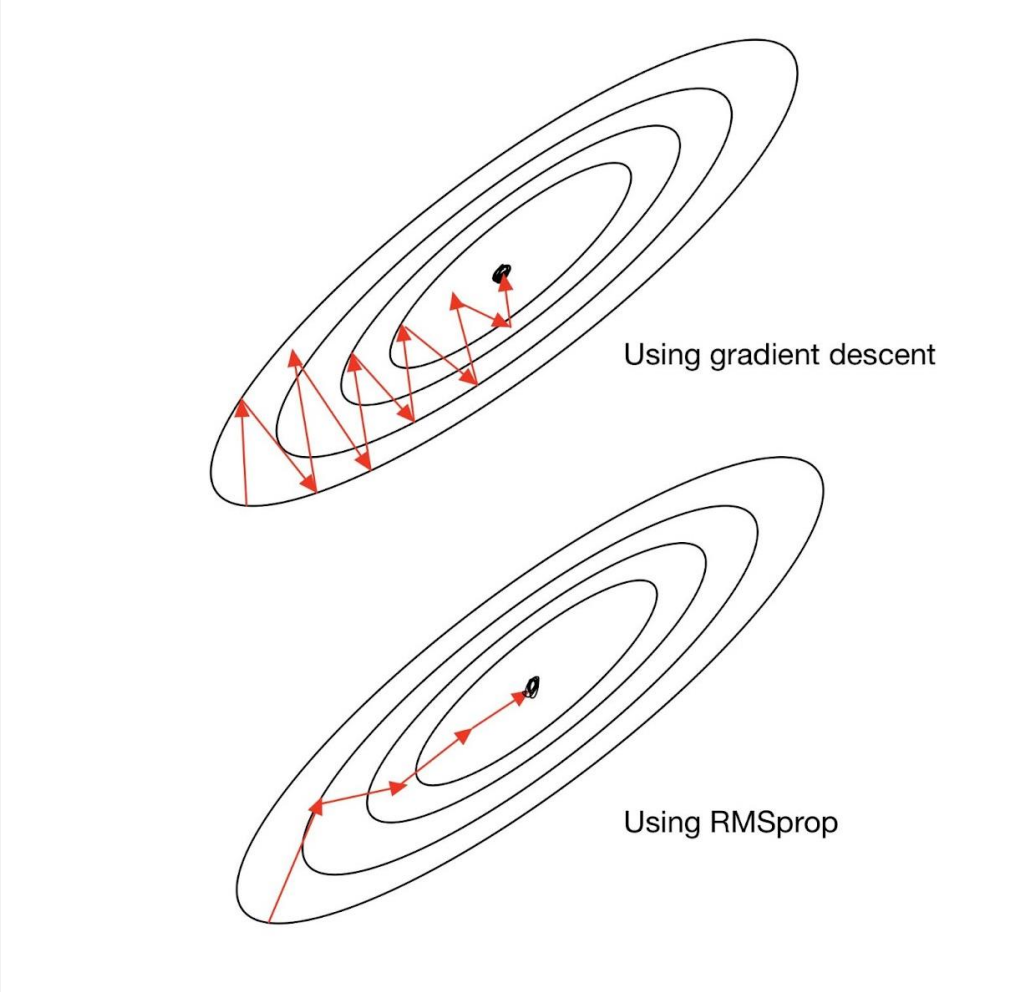
출처 https://amber-chaeeunk.tistory.com/23

RMSProp

Root Mean Square Propatation의 약자

Adagrad의 계산식에

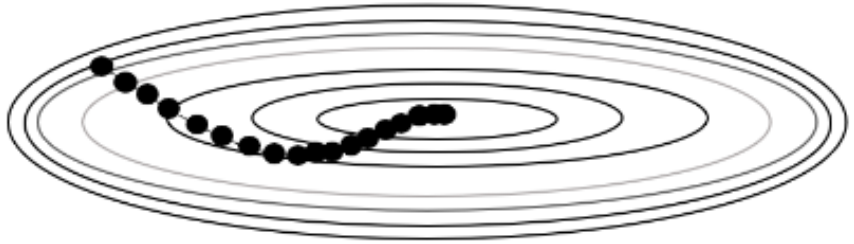
지수 가중 이동 평균(Exponentially weighted average)를 사용하여 최신 기울기들이 더 크게 반영되도록 함



Adam

Adam(Adaptive Moment Estimation) 은
RMSProp과 Momentum 방식을 융합

Momentum과 유사하게
지금까지 계산한 기울기의 지수 평균을 저장하며,
RMSProp과 유사하게
기울기의 제곱값의 지수 평균을 저장

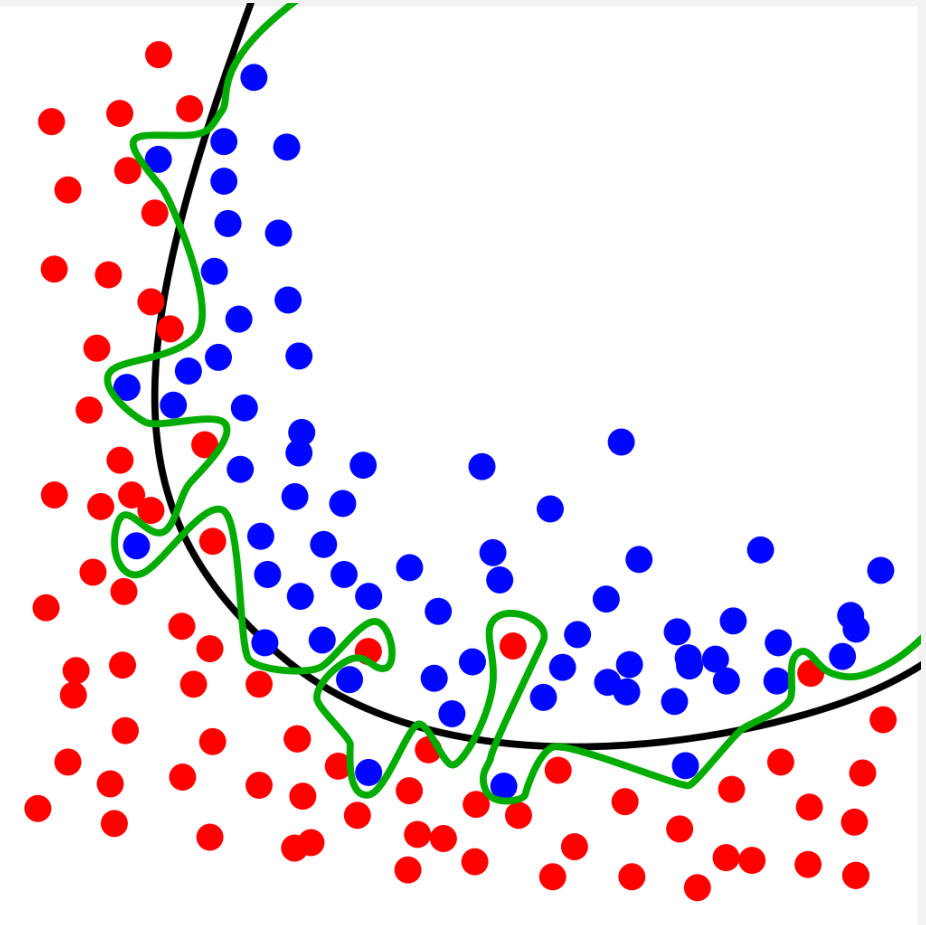


Adam

출처 https://amber-chaeeunk.tistory.com/23

Overfitting과 Early Stopping

과적합 (Overfitting)



출처 https://ko.wikipedia.org/wiki/%EA%B3%BC%EC%A0%81%ED%95%A9

Figure 5. overfitting

- 과적합은 모델이 학습 데이터에 너무 과도하게 적합화되어, 새로운 데이터에 대한 예측 성능이 저하되는 현상

- 모델이 학습 데이터에만 맞추어져 새로운 데이터에 대해 일반화(generalization) 능력이 부족하게 되는 것

초록색 선은 과적합된 모델을, 검은색 선은 일반 모델

Early Stopping

- 과적합을 방지하기 위한 방법 중 하나
- 학습 중 모델의 성능을 모니터링하다가 학습 데이터에 대한 성능은 계속해서 향상되지만 검증 데이터에 대한 성능은 향상되지 않거나 감소하는 지점에서 학습을 조기 종료하는 것
- 일반적으로, 학습 데이터와 검증 데이터를 나누어 학습하면서 적용됨

```
#3. 컴파일, 훈련
model.compile(loss='mse', optimizer='adam')

from keras.callbacks import EarlyStopping
earlyStopping = EarlyStopping(monitor = 'val_loss', patience=100, mode='min', verbose=1,
                              restore_best_weights=True)
                              #"restore_best_weight=False"가 Default => 최적의 "Weight"를 찾기위해 "True"로 해주어야 함

start_time =time.time()
hist = model.fit(x_train, y_train, epochs=1000, batch_size=1,
                 validation_split=0.2,
                 callbacks=[earlyStopping],
                 verbose=1)
end_time =time.time()
```

Figure 6. early stopping 코드

실 습

1. Overfitting
2. Early Stopping
3. Save model
4. Save weight
5. Model Check Point
6. Pandas
7. Heatmap (matplotlib + seaborn)

Day03. 인공지능 Study

1. 인공지능 개념 정리 – 머신러닝, 딥러닝
2. 퍼셉트론 (Perceptron)
3. 다층 퍼셉트론 (Multi-Layer Perceptron: MLP)
4. 옵티마이저 (Optimizer)
5. 학습률 (learning rate)
6. 경사하강법 (Gradient Descent)
7. 손실함수 (Loss Function)
8. 활성화 함수 (Activation Function) – Sigmoid, ReLU, Softmax

9. 회귀분석

10. 결정계수 R2 score

11. 분류분석

12. 원 핫 인코딩 (One Hot Encoding)

13. 난수값 (random_state)

14. 정확도 accuracy score

15. 과적합 (overfitting)

수고하셨습니다.