

머신러닝

Learning Rate, Overfitting, Regularization

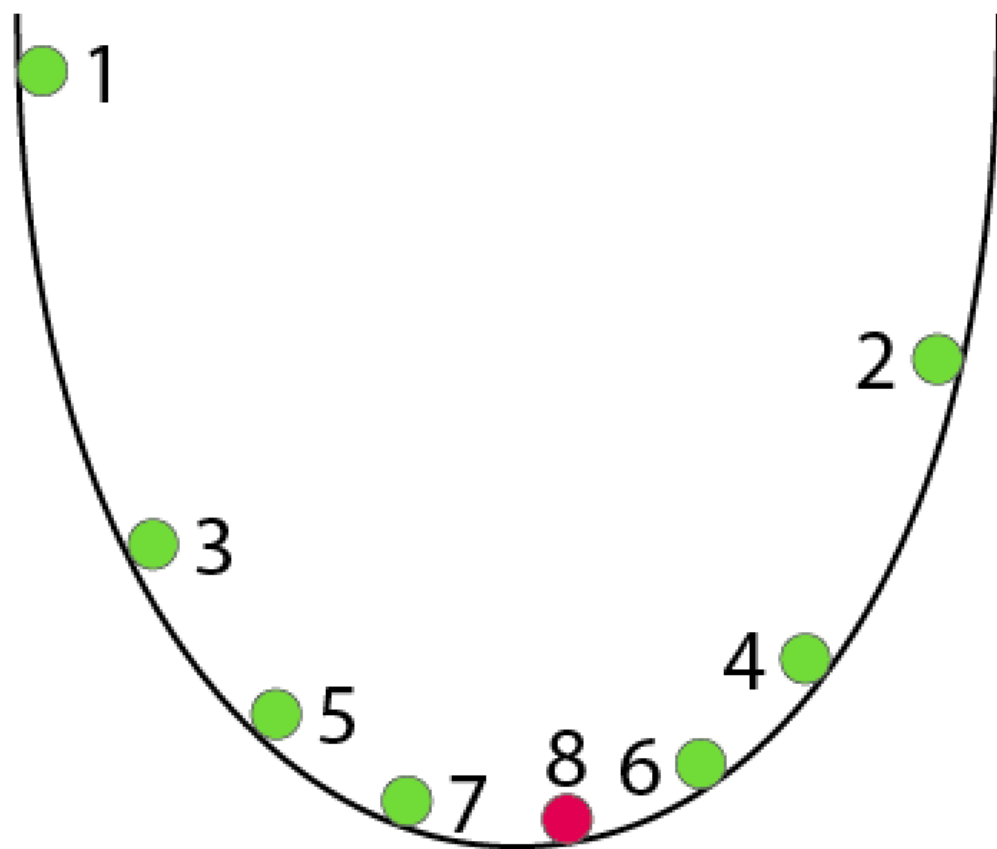
이승우

1. Gradient Descent란 무엇인가?

최적화 함수의 한 종류
(Back-Propagation도 있어요!)

경사를 하강 시켜서 최적의 값을 찾는다

공을 떨어워서 공이 멈춘 곳이 바로 그 값!



근데 문제가 하나 있어요

이게 자칫하면 멈추질 않아요

이상한 곳으로 가거나 느려요



이렇게 된 이상 이상한곳으로 간다!

이런 현상을 오버슈팅 이라고 하는데
(~~오버워치~~ 아닙니다)

이를 위한 해결책은...

Learning Rate을 잘 설정해서

너 여기서 여기까지 몇걸음에 가~

정확한 위치에 **안착**할 수 있도록 해줍니다.

근데 잘 설정하는 방법은 **없어요(...)**

과적합

데이터를 분류하고 보니!

말그대로

“너무 심하게 적합된” 것.

마치 간단하게 한줄만 그으면 되는걸

뱀이 지나가듯이 이리꼬고 저리꼬는거...

뱀이 지나가지 못하도록 데이터를 간단하게

해보자^[1]

[1]: 정칙화(**Regularization**) 이라고 합니다, 데이터를 간단하게 만들어서 과적합에 빠지지 못하도록 합니다

근데 근데 근데!

데이터가 너무 커!

뱀 자체가 못지나가!

그럼 이제 어떻게 해야하지!

아! 그럼 되겠다!

데이터를 쪼개서^[1]

뱀이 중간에 눈치 못채도록 해야겠다!

[1]: **Validation** 이라고 합니다, 학습용과
검증용을 나누어서 사용합니다.

아! 그래도 뱀이 눈치챌수 있으니까
매 학습 때마다 전혀 보지도 못한 데이터^[1]를
주자!

[1]: Cross-Validation 이라고 합니다.

각 학습때마다 학습-검증 값을 무작위로 선택합니다

이렇게만 해주면 뱀은 우리가 원하는 길로
지나갈 겁니다