



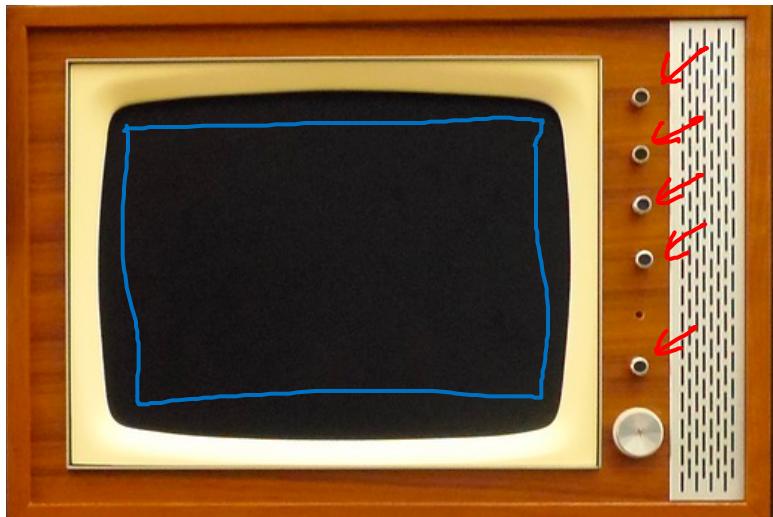
deeplearning.ai

Introduction to ML strategy

Orthogonalization

딥러닝의 challenge 중 하나는 시도해보고 바꿔볼게 너무 많다는 점.
똑똑한 애들은 어떤 결과를 위해서 무엇을 튜닝해야 하는지 잘 알고 있다.
이것이 바로 orthogonalization이라는 프로세스이다.

TV tuning example



Orthogonalization

하나의 스위치를 돌렸을 때 저런 것들이 $0.1, 0.3, -1.7, \dots$
이런 식으로 바뀐다면 티비 화면을 보정하기 거의 불가능할
것이다. 각 스위치가 각 기능을 수행하는게 낫다.

$$\begin{aligned}
 & 0.1x \quad \boxed{\uparrow} \\
 & + 0.3x \quad \boxed{\leftarrow} \\
 & - 1.7x \quad \boxed{-} \\
 & + 0.8x \quad \boxed{\leftarrow \rightarrow} \\
 & + \dots \quad \boxed{-}
 \end{aligned}$$



$$\begin{aligned}
 & \rightarrow \underline{0.3 \times \text{angle}} - 0.8 \times \text{speed} \\
 & \rightarrow 2 \times \text{angle} + 0.9 \times \text{speed} .
 \end{aligned}$$

speed
 angle
 0

Chain of assumptions in ML

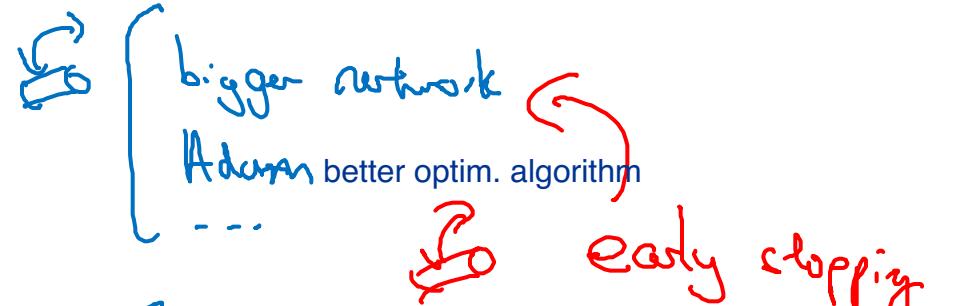
supervised learning 시스템에서 돌려야 할 스위치는 아래처럼 4가지

각 단계에서의 knob은 아래와 같다.

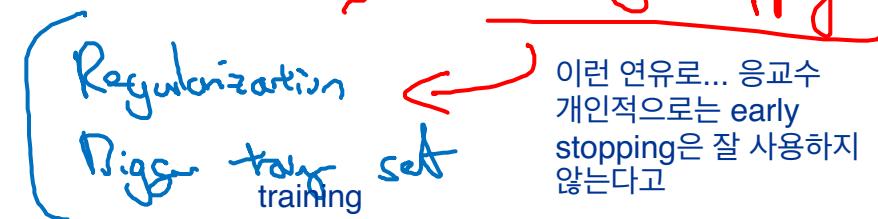
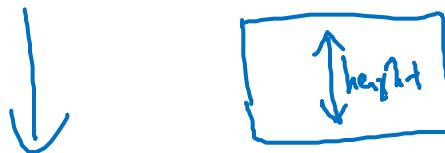
→ Fit training set well on cost function



(\approx human-level performance)



→ Fit dev set well on cost function



→ Fit test set well on cost function



Bigger dev set

→ Performs well in real world

(Happy cat pic off users.)

Change dev set or
cost function

왜냐하면 어떤 cost function으로는 test set에 잘 작동한다는게 꼭 real world에서도 잘 작동한다는걸 의미하지는 않는다. 생각처럼 안된다는건 사용한 cost function이 right thing을 제대로 측정하지 못하고 있거나, 아니면 dev - test set의 distribution이 적절하게 세팅되지 않았다는 의미이다.

Andrew Ng