



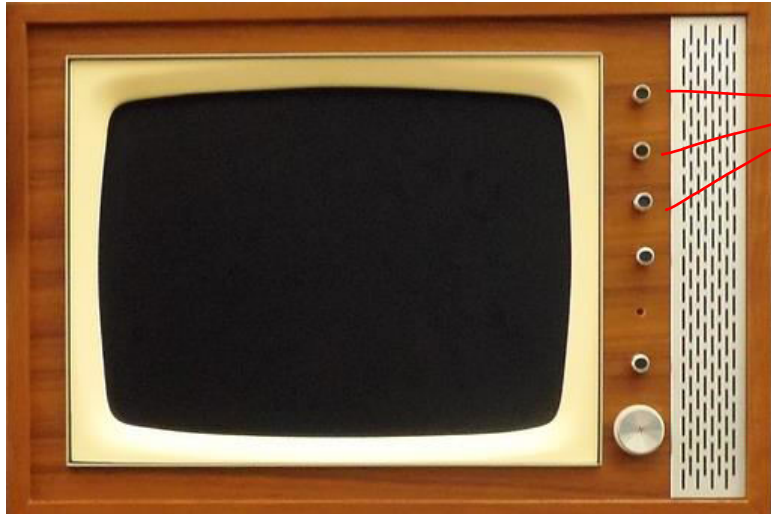
deeplearning.ai

Introduction to ML strategy

Orthogonalization

- What idea to Implement, that will lead to what gain?

TV tuning example



Multiple knobs
to Adjust the TV
picture Quality

1st knob →

2nd →

3rd → contrast

4th → Black white
v/s color
etc.

- Suppose, the TV picture Quality was determined by a preset eqn
$$= 0.1(\text{vertical width}) + 0.3(\text{horizontal}) - 1.7(\text{contrast}) + \dots$$
- Basically, you have 1 knob that does all of this, that would be bad as you can't figure out what is causing the bad image!
- orthogonalization - Have multiple knobs that handle 1 aspect each

Car



Car has 3
components for its
movement

- Steering
- Acceleration
- Braking

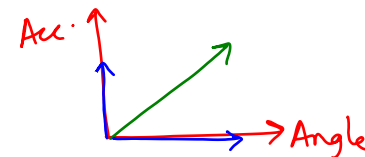
If you had 2 knobs

① $0.3 \times (\text{steering Angle}) - 0.8 (\text{Acc})$

② $2 \times (\text{steering Angle}) + 0.7 (\text{Brakes})$

- Then it would be much harder to control car motion, compared to 1 knob per component

orthogonal controls
↳ Independent of each other!



■ = orthogonal

■ = Non orthogonal

orthogonal =
90° to each
other

Chain of assumptions in ML

Fit training set well on cost function



(criteria may be to perform as good as a human)

Fit dev set well on cost function



- knobs in case of DNN

① Bigger N/w

② Adam v/s momentum

....

Fit test set well on cost function



Performs well in real world

If model does well on training + dev set, but not test set, knob = get more dev set data

- If DNN does well on training set, but not dev set, it may mean you have overfit \Rightarrow There are knobs specific to this

① Regularization
 \hookrightarrow Dropout, early stopping

② More data etc.