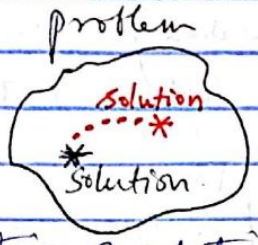
Feb 14th, 2017 All ANNs we have learned so far are useless if we have small datasets or even worse, we have none.

Can we learn from scratch?

↳ Mechanism? Learning by doing.

Online learning.

Online learning is necessary for several reasons:
* data not available.
* non-stationary.

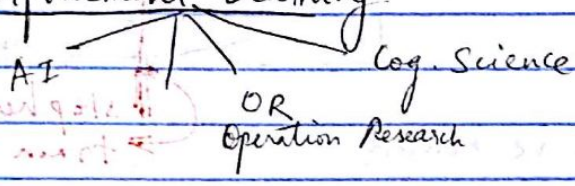


Stationary: Solution is fixed.

Learning by doing
- No teacher
- But there is indirect feedback. (either punishment or reward.)
- the possibility to get permanent / constant / continuous feedback.

Location of solution in feature space is fixed. "stationary"
non stationary: solution moves.

Reinforcement Learning

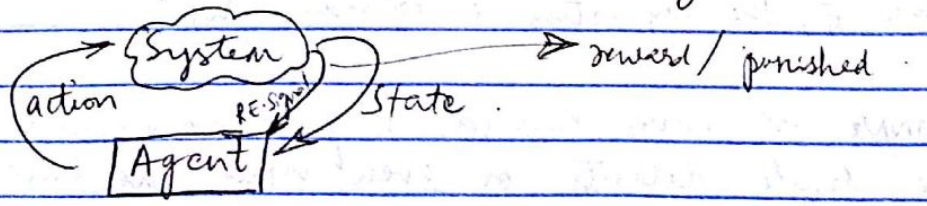


RL has nothing to do with NN.

Typical Applications
① Robotics - elevators control
- Robo-soccer
② Board games - Backgammon
- Checkers
- Chess
③ Scheduling - Dynamic channel allocation
- inventory problems.

Smart: find a trade off between exploitation & exploration.

Idea of RL (Reinforcement Learning)



Taking actions and observering reward/punishment.
↳ take the right (exploitation)
↳ or experiment (exploitation).

RL math Model

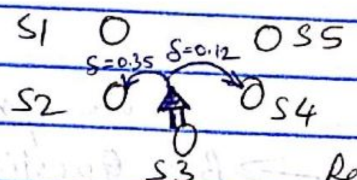
- Stochastic/non-stationary : uncertainity
- decision making under uncertain conditions.
- weakly supervised : maximize reward.

] Markov Decision Process (MDP)

MDP

- ↳ Set of states S
- ↳ Set of actions A
- ↳ Transition probability δ
- ↳ Reward function R

Environment : Set of states



taking actions : $\delta(S, a, S')$

Reward : $R(S, a)$

$R(s, a) = -1$ (with $\delta = 0.5$)
$= +10$ (with $\delta = 0.35$)
$= +20$ (with $\delta = 0.15$)

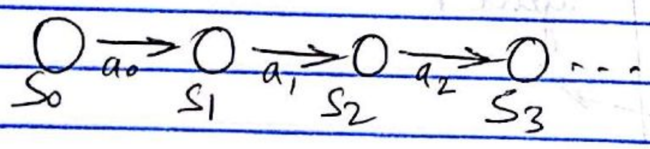
Cardinality of $A <<$ Cardinality of S

actions

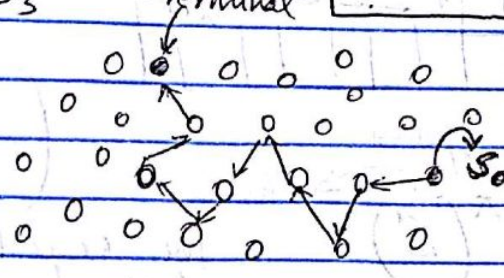
States	a_1	a_2	a_3	a_i
S_1	0.1	0.	0.08	0.51
S_2	0.4	0.69	0.17	0.15
S_3	0.2	0.4	0.14	
⋮				

(it will be a long matrix)

MDP Model : Trajectory



Finding trajectory = learning



Any scheme working based on reward has to answer this question : immediate or long-term ?

"greedy" approach.

Another question : Are early rewards more important than later ones? . (Early ones)

↳ discount the rewards?

Return the function (sum of rewards)

① Finite horizon

$$\text{return} = \sum_i R(s_i, a_i)$$

② Infinite horizon

(2a) discounted, $\text{return} = \sum_{i=0}^{\infty} \gamma^i R(s_i, a_i)$ $\quad 0 < \gamma < 1$

(2b) undiscounted,

$$\text{return} = \frac{1}{N} \sum_{i=0}^{N-1} R(s_i, a_i)$$

↳ one episode with N iterations/observations.

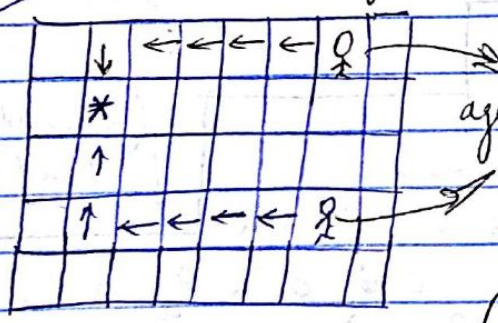
⟹ Big question: Can we observe the system?

↳ Yes, fully observable

↳ No, partially observable (maybe we have some hidden variables)

The goal of R.L. is to find a policy π that can guide the agent along an optimal trajectory.

Grid world problem:



$V^{\pi}(s)$ = the expected return starting at state s following π.

(π: policy; s: state; V: value)

(find the value of which state under the conditions of policy π)

$Q^{\pi}(s, a)$ = the expected return starting at state s with action a and then following π.

(π: policy; s: state; a: action; Q: substitution (the value))