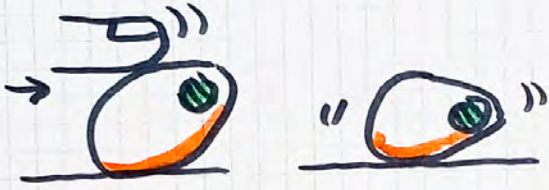
COGS 300 Learning 02 Mar 19/26 ①

Warm up: weights + stability.

Draw a shape:

add a weight

Disturb it:

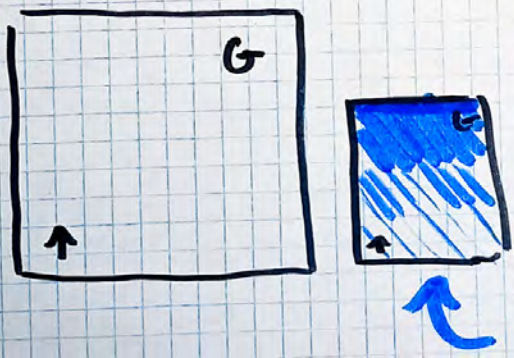


find stable point

Repeat: try different shapes + weights.

etc.

genetic programming evolutionary alg.



N	S	E	W
1/4	1/4	1/4	1/4
91/100	3.3/100	3.3/100	3.3/100

★ What are the traps for this specific goal + env.

if sensor = wall:

N S E W

else:

N	S	E	W
91/100	3.3/100	3.3/100	3.3/100

★ How to reweight if seeing a wall ?

if s=w:

N	S	E	W
0	0	0	1

✗

if s=w:

N	S	E	W
0	1/3	1/3	1/3

if s=w and last = X: $X \in \{N, S, E, W\}$

N	S	E	W
=	=	=	=

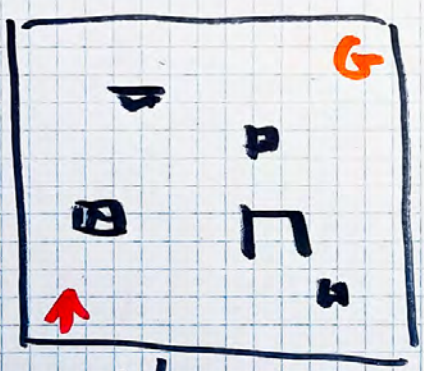
eg. if s=w and last = N:

N	S	E	W
0	1	0	0

✓

1/100	97/100	1/100	1/100

✓✓



$W_N = W_N \cdot \frac{1}{2}$

re-normalize

★ what is punishable? rewardable

★ what is punishment? reward

learning rate

Learning 02

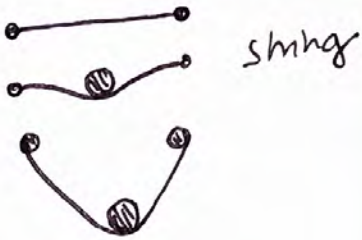
warm up: weight.

draw an oval

stabilize

add weight

find stable spot



string

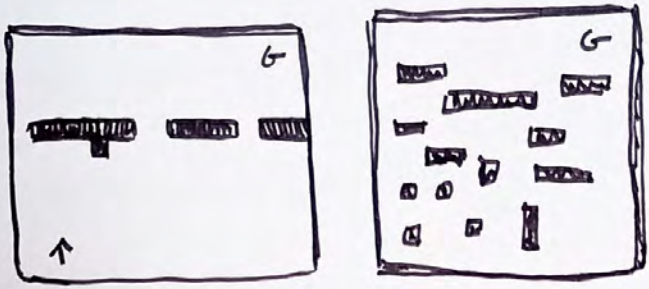
stable or not?

Last time: weighting decisions / actions.

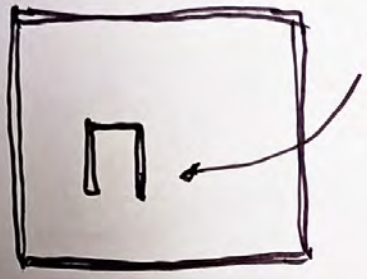
	N	S	E	W	
gene	1/4	1/4	1/4	1/4	trait: random
gene	91/100	3.5/100	3.5/100	33/100	trait: mostly N

probability density function for gene.

Note how this save is "good" at solving minor mazes:

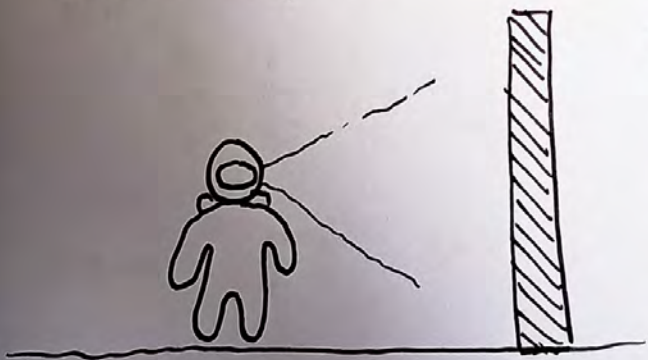


★ what traps the save?



problem: observability.

As you know, agents don't know where they are.



like being at the bottom of the ocean ... and air is running out!

even if you had a grid... try to remember!

we're keeping probabilities b/c:

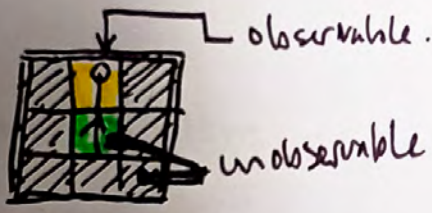
1. more realistic model of noise.
2. solves problems of pure determinism (traps)

observability of global state is very hard! Realistically:

agent doesn't know x, y or d.r.

but it does know sensor + action.

start with just sensor.



if sensor = 1:

N	S	E	W
0	1/3	1/3	1/3

else:

N	S	E	W
91/100	3.3/100	3.3/100	3.3/100

now  isn't a trap!

if: sensor = 1 and last = N:

N	S	E	W
0	1	0	0

if sensor = 1 and last = E:

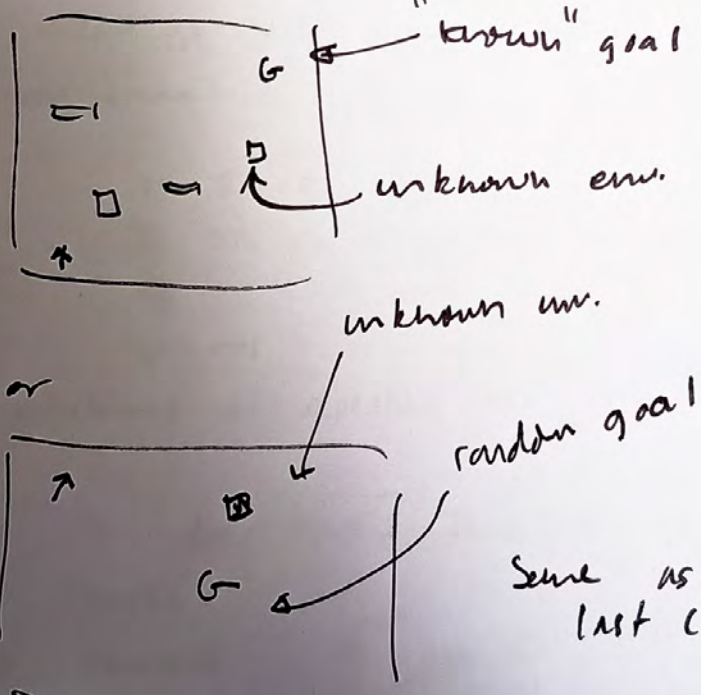
N	S	E	W
0	0	0	1

etc.

if sensor = 0 and last = N:

→ base policy.

imagine setting up exp. for sen:



But we want reward / punishment onboard!
Let's punish a policy! (gene, action...)

if last = N and s = 1:

N	S	E	W
w_N	w_S	w_E	w_W

↓

w_N - punishment (min of zero)

If punishment is severe:

N	S	E	W
0	1/3	1/3	1/3

We never go north again... problem!

If punishment is light:

N	S	E	W
[illegible]	~1/4	~1/4	~1/4

1/4 - 0.0001

Hi likelihood of repeating bad actions.

* How to solve this? Think:
  1. State
  2. reward. Balance vs. this.