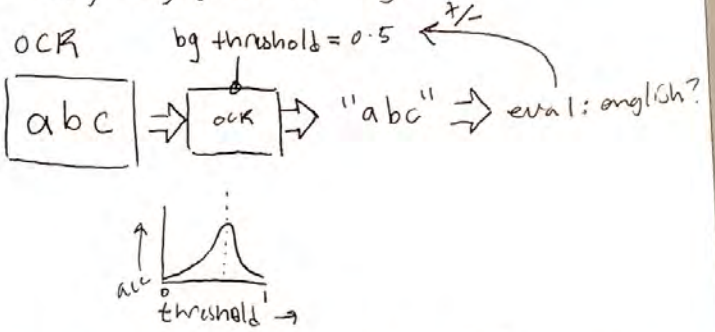
COGS 300 Distribution 04 Nov 20/25

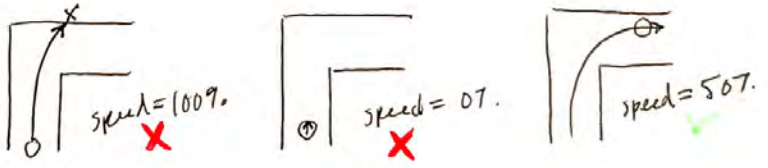
Last two classes, we've discussed reinforcement learning. Now, we will ask: how does a multi-agent system or network learn?

One evaluative "function" is death. Evolutionary algorithms leverage this concept: each "agent" has a set of parameters that randomly vary & weighted by success.




cutting edge AI agents basically find a way to form the eval to be in text.

An example might be learning to take the right speed at a turn:




The question is how to apply a reward fn.

Let's sketch a reasonable approach:

learn ②

Raw: $u_L$ $u_M$ $u_R$ $E_L$ $E_R$ | $O_L$ $O_R$

↓ filter ↓ filter ↓ filter ↓ speed ↓ speed ↓ ↓

features → $dist_L$ $dist_M$ $dist_R$ $sp_L$ $sp_R$ $O_L$ $O_R$

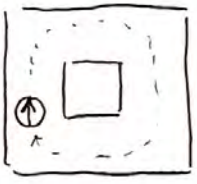
① How do you detect running into a wall?

② How do you punish it?

③ How do you reward not running into a wall?

Ex. if $O_L > 100$ and $sp_L < 10$ → -100

④ How do you actually do this irL?



ex.

But how do you evaluate the evaluator?

Back propagation

Let's go back to the 1950s.

The history of CS really is the history of trying to make thinking machines.

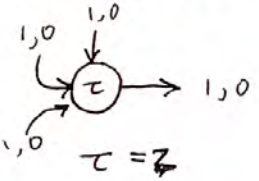
The architecture of modern ML was designed in the 1950s. Backprop. in 1970s.

McCulloch + Pitts developed a Calculus of a neuron. in 1943.

Frank Rosenblatt built a machine to actually calculate in 1957.

This is the perceptron.

Idea: using circuits, implement the function of a neuron.




$\tau = 2$

This should look like an extension of a logic gate.

| a | b | c | out = a+b+c ≥ τ ? |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

$p_{00}$ $p_{01}$ $p_{02}$ ⋯ $p_{16}$ $\quad out = \sum p_{ij} \geq \tau$ ?

$\tau = 12$

vs

both AI?

activation

×

weighted mask

★ How do you design the weights so that:

correct ✓

correct ✓

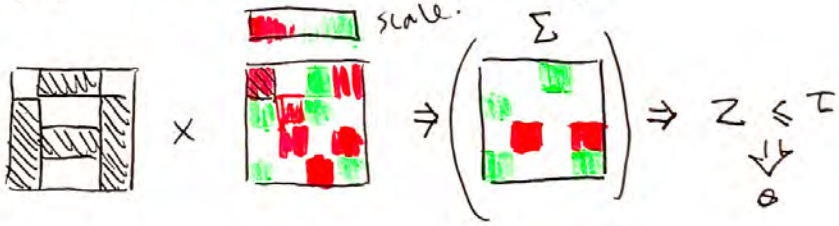
} only two correct.

incorrect ✗

etc.

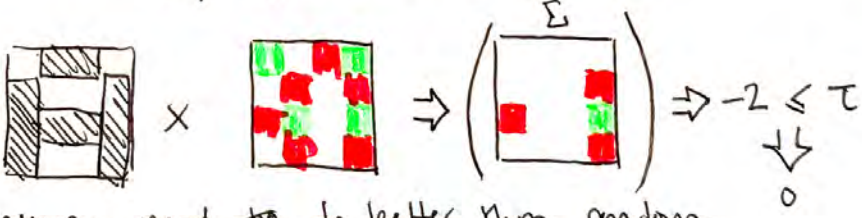
any other example incorrect.

Backprop: How to do this automatically?




but we know should be 1.

Try again! New weights:




hmm... need to do better than random.

★ How can we evaluate our weights?

loss: how far are we from the goal?

computing power simply wasn't enough until
~ the 2010s to make this

Deep Blue GOFAI IBM.

Watson NLP ML

currently, best approaches mix. GOFAI + ML.

☆ Design an architecture for your robot to leverage AI.

→ which parts can be ML?

→ which parts can be GOFAI?

→ which parts need LLM eval?

↳ how to construct it?

CGGS 300 Distribution 04 Nov 20/25

Warm up: Draw isomorphisms of a random walk.

Eg.  

etc.

random walk — random walk's angle

 ...

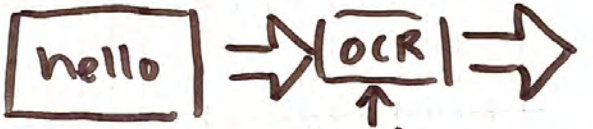
random walk in pixel space.

Tonight - sun: East Van culture crawl
→ Van Hack Space on Venebles.

Non-mathy reward functions?

Optical character recognition (OCR)

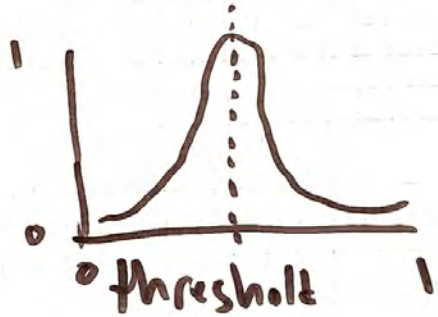
Img → text

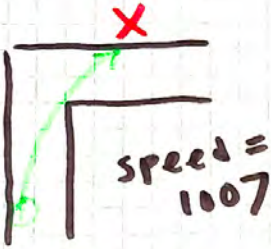



background threshold

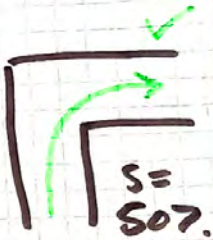
eval if english = good!

0.5







action

0-255
0-100%.

state

$U_L$ $U_m$ $U_R$ $E_L$ $E_R$ ..... $M_L$ $M_R$

$U_L$ ↑ Dist L | $U_m$ Dist m | $U_R$ Dist R | $E_L$ Speed L | $E_R$ Speed R | $M_L$ ↓ out L | $M_R$ ↓ out R

① Detect running into a wall

② Reward / punish?

③ can you train the robot autonomously?

① No change to dist.

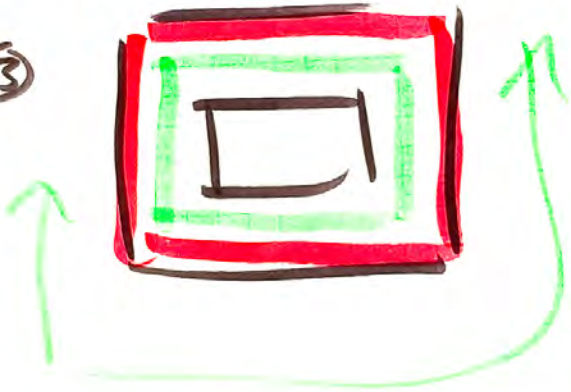
If encoder < t → not moving

②

int total_dist

loop:
  td += dist
  reward(total_dist)

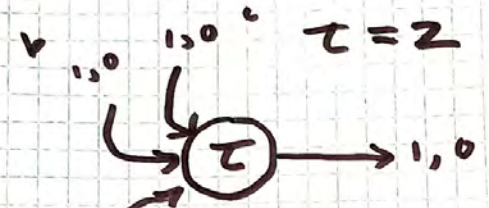
③



McCulloch + Pitts.

Frank Rosenblatt 1957

perceptron



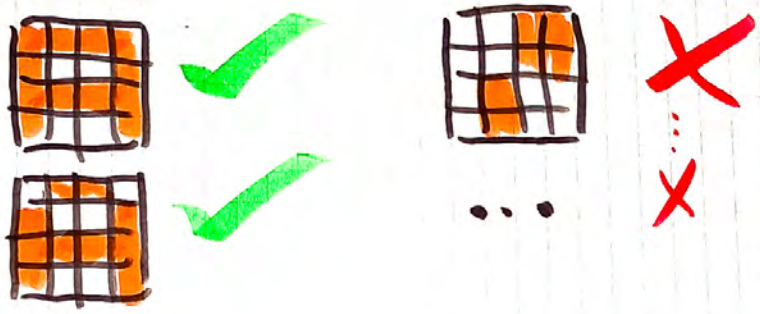

| a | b | c | out = a+b+c ≥ τ ? |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |


| $P_{00}$ | $P_{10}$ | $P_{20}$ | ⋯ | $P_{01}$ | ⋯ | $P_{ij}$ |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | | 0 | | |

$$out = \sum P_{ij} \geq \tau$$

$$\tau = 12$$

weighted mask



come up with a mask that correctly classify

-1, 1


loss